PATENT ABSTRACTS OF JAPAN

(11)Publication number : 10-208026
(43)Date of publication of application : 07.08.1998

(51)Int.Cl. G06T 1/00
G06T 7/00
G09C 5/00
H04N 1/387
H04N 1/44

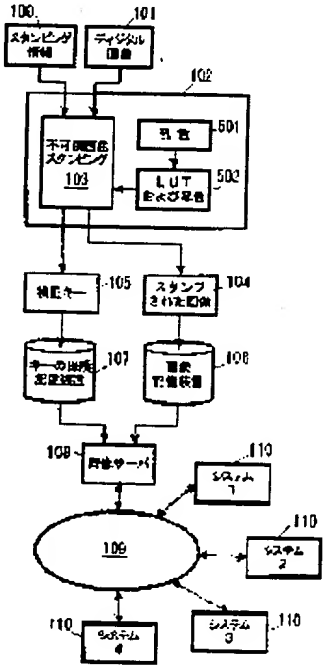
(21)Application number : 09-336757
(22)Date of filing : 08.12.1997

(71)Applicant : INTERNATL BUSINESS MACH CORP <IBM>
(72)Inventor : MINTZER FREDERICK COLE
YEUNG MINERVA MING-YEE

(30)Priority
Priority number : 97 780484 Priority date : 08.01.1997 Priority country : US

(54) PICTURE VERIFICATION DEVICE

(57)Abstract:
PROBLEM TO BE SOLVED: To verify that contents of a picture are not changed after being stamped by discriminating alteration of the stamped picture at the time extracted stamping information is unmatching with source stamping information.
SOLUTION: A picture server 108 verifies the integrity of picture data stored in a picture archive 106. That is, a source picture 104 stamped from the archive is selected, and each information is read into a calculation device after a verification key 105 corresponding to a picture stamped from a key protection storage device 107 is acquired. In the verification processing, the stamped source picture and the verification key are processed to extract stamping information embedded in the picture. Next, extracted stamping information and stamping information 100 known by the server are compared with each other. If they coincide with each other, it is judged that this picture is not changed after being stamped. This verification result is used to confirm the integrity of contents, and a proper processing is performed if it is verified that contents have been changed.



LEGAL STATUS

[Date of request for examination] 16.10.1998

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

(19)日本国特許庁（ＪＰ）　(12)公開特許公報（Ａ）　(11)特許出願公開番号

特開平10－208026

(43)公開日　平成10年(1998)８月７日

(51)Int.Cl.[6]		識別記号	FI		
Ｇ０６Ｔ	1/00		Ｇ０６Ｆ	15/62	３８０
	7/00		Ｇ０９Ｃ	5/00	
Ｇ０９Ｃ	5/00		Ｈ０４Ｎ	1/387	
Ｈ０４Ｎ	1/387			1/44	
	1/44		Ｇ０６Ｆ	15/70	４６０Ａ

審査請求　未請求　請求項の数28　ＯＬ　（全 23 頁）

(21)出願番号　特願平9－336757

(22)出願日　平成９年(1997)12月８日

(31)優先権主張番号　０８／７８０４８４

(32)優先日　1997年１月８日

(33)優先権主張国　米国（ＵＳ）

(71)出願人　390009531
インターナショナル・ビジネス・マシーンズ・コーポレイション
ＩＮＴＥＲＮＡＴＩＯＮＡＬ　ＢＵＳＩＮＥＳＳ　ＭＡＳＣＨＩＮＥＳ　ＣＯＲＰＯＲＡＴＩＯＮ
アメリカ合衆国10504、ニューヨーク州　アーモンク　（番地なし）

(72)発明者　フレデリック・コール・ミンツァー
アメリカ合衆国10588　ニューヨーク州シュラブ・オーク　オリオール・コート　3798

(74)代理人　弁理士　坂口　博　（外１名）

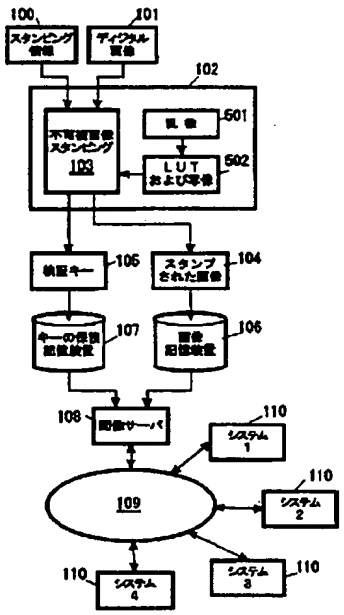
最終頁に続く

(54)【発明の名称】　画像検証装置

(57)【要約】

【課題】　本発明の目的は、以前に画像の内容がスタンプされた時以降に画像の内容が変更されていないことをすばやく検証するシステムを提供することである。

【解決手段】　本発明のシステムは、スタンピング情報をソース画像に埋め込み、検証キーを作るスタンピング処理と、検証キーに基づいてスタンプされたソース画像からスタンピング情報を抽出する検証処理からなる。さらに、検証処理では、画像自体が作られ、この画像から、視覚的にまたはコンピュータその他の表示装置の使用によって、検証を簡単に判定できる。検証処理では、画像の変更を検出でき、その位置を限定することができる。画像スタンピング処理には、さらに、誤差拡散処理が含まれ、その結果、元の画像にスタンピング情報を組み合わせた影響を、簡単には知覚できなくなる。画像は、悪意のある操作から保護され、所有権は、画像内容の完全性を維持することによって保護される。



1

【特許請求の範囲】

【請求項１】（ａ）少なくとも１つの参照テーブル（ＬＵＴ）に対応する種値及び写像関数から検証キーを形成することと、

（ｂ）それぞれの抽出値を形成するため、写像関数と少なくとも１つのＬＵＴとに従って、複数のソース画像値のそれぞれを写像することと、

（ｃ）スタンピング情報のうちの対応する部分と同等の変更された抽出値を有する、それぞれの変更されたソース画像値を作るため、スタンピング情報のうちの対応する部分と一致しない抽出値を有する、複数のソース画像値のうちの選択されたソース画像値を変更することとによって、スタンプされた画像を作るための画像スタンピング手段と、

スタンプされた画像を受け取り、写像関数に従ってスタンピング情報を抽出するために検証キーを使用し、抽出されたスタンピング情報を提供するための、画像抽出手段と、

スタンピング情報と抽出されたスタンピング情報とが実質的に同等でない場合に、スタンプされた画像への変造があることの信号を送出するための信号送出手段とを含む、複数のソース画像値を有するソース画像を、スタンピング情報を用いてスタンプし、検証するための装置。

【請求項２】検証キーが、種値を含み、画像スタンピング手段が、さらに、乱数ジェネレータ及び種値を使用して少なくとも１つのＬＵＴの各項目を生成するためのテーブル生成手段を含むことを特徴とする、請求項１に記載の画像検証装置。

【請求項３】スタンピング情報が、透かし画像であることを特徴とする、請求項１に記載の画像検証装置。

【請求項４】検証キーとスタンプされた画像を組み合わせるための組み合わせ手段をさらに含み、

画像抽出手段が、スタンプされた画像から検証キーを抽出する手段をさらに含むことを特徴とする、請求項１に記載の画像検証装置。

【請求項５】ソース画像が、複数のソース画素値から形成され、透かし画像が、複数の透かし画像画素値から形成されることを特徴とする、請求項３に記載の画像検証装置。

【請求項６】画像スタンピング手段が、さらに、複数のソース画素値のそれぞれをアドレッシングする順序を決定するための手段と誤差拡散手段とを含み、

誤差拡散手段が、

１ソース画素と、ソース画像内の、誤差値を含む対応する修正画素とを識別するための手段と、

誤差値を含む修正画素の値と、スタンプされたソース画像内の対応する画素の値との間の差として定義される修正画素誤差の量を決定するための手段と、

隣接画素値の調整の総量が修正画素誤差と同等になるように、連続する隣接ソース画素値を、識別されたソース画素から離れるにつれて量が減少する修正画素誤差を用いて調整する手段を提供することによって、修正画素誤差を誤差拡散するための手段とを含むことを特徴とする、請求項５に記載の画像検証装置。

2

【請求項７】少なくとも１つの参照テーブル（ＬＵＴ）に対応する種値及び写像関数から検証キーを形成するための形成手段と、

それぞれの抽出値を形成するため、写像関数及び少なくとも１つのＬＵＴに従って、複数のソース画像値のそれぞれを写像するための写像手段と、

スタンピング情報のうちの対応する部分と同等の変更された抽出値を有する、それぞれの変更されたソース画像値を作るため、スタンピング情報のうちの対応する部分と一致しない抽出値を有する、複数のソース画像値のうちの選択されたソース画像値を変更するための画素変更手段とを含む、複数のソース画像値を有するソース画像を、スタンピング情報を用いてスタンプするためのシステム。

【請求項８】複数のソース画像値が、複数のソース画素値であり、スタンピング情報が、複数の透かし画像画素値から形成される透かし画像であることを特徴とする、請求項７に記載の画像スタンピング・システム。

【請求項９】信号送出手段が、さらに、

スタンプされる情報及び抽出されたスタンピング情報を記憶するための手段と、

比較情報を作るため、スタンプされる情報と抽出されたスタンプされた情報を比較するための手段と、

抽出されたソース画像に対する変造を比較情報から検出し、抽出されたソース画像のうち、変造に対応する少なくとも１つの領域を示すための手段とを含む、請求項１に記載の画像検証装置。

【請求項１０】スタンプされた画像が、第１メモリに記憶された複数のスタンプされた画像のうちの１つであり、複数のスタンプされた画像のそれぞれが、対応する検証キーを有し、検証キーのそれぞれが、第２メモリに記憶された複数のキーのうちの１つであることを特徴とする、請求項１に記載の画像検証装置。

【請求項１１】第１メモリ内の複数のスタンプされた画像からスタンプされた画像を選択するための手段と、

第２メモリに記憶された複数のキーから、スタンプされた画像に対応する検証キーを取り出すための手段とをさらに含む、請求項１０に記載の画像検証装置。

【請求項１２】さらに、第１メモリ内の複数のスタンプされた画像のそれぞれの完全性を検証するための画像完全性検証手段を含み、画像完全性検証手段が、（１）複数のスタンプされた画像のうちのそれぞれ及び対応する検証キーを選択することと、（２）対応する検証キーに基づいてスタンプされた画像からスタンピング情報を抽出することと、（３）検証キーを用いてスタンピング情報を検証することによってスタンプされた画像の完全性

を検証することとを反復することを特徴とする、請求項１１に記載の画像検証装置。

【請求項１３】（ａ）少なくとも１つの参照テーブル（ＬＵＴ）に対応する種値及び写像関数から検証キーを形成することと、

（ｂ）それぞれの抽出値を形成するため、写像関数と少なくとも１つのＬＵＴとに従って、複数の周波数領域係数値のそれぞれを写像することと、

（ｃ）スタンピング情報のうちの対応する部分と同等の変更された抽出値を有する、それぞれの変更された周波数領域係数値を作るため、スタンピング情報のうちの対応する部分と一致しない抽出値を有する、周波数領域係数値のうちの選択された周波数領域係数値を変更することとによって、スタンプされた画像を作るための画像スタンピング手段と、

スタンプされた画像を受け取り、写像関数及びＬＵＴに従って複数の周波数領域係数値の一部からスタンピング情報を抽出するために検証キーを使用し、抽出されたスタンピング情報を提供するための、画像抽出手段と、

スタンピング情報と抽出されたスタンピング情報とが実質的に同等でない場合に、抽出されたソース画像への変造があることの信号を送出するための信号送出手段とを含む、複数の周波数領域係数値によって表されるソース画像を、スタンピング情報を用いてスタンプし、検証するためのシステム。

【請求項１４】複数の周波数領域係数値が、複数のソース画像画素値によって表されるソース画像に対する周波数領域変換演算によって形成されることを特徴とする、請求項１３に記載の画像検証システム。

【請求項１５】ソース画像画素値に対する変換演算が、離散コサイン変換（ＤＣＴ）であり、複数の周波数領域係数値の一部が、ＤＣＴ変換された複数のソース画像画素値の直流係数を含むことを特徴とする、請求項１４に記載の画像検証システム。

【請求項１６】スタンピング情報が、透かし画像であることを特徴とする、請求項１３に記載の画像検証システム。

【請求項１７】少なくとも１つの参照テーブル（ＬＵＴ）に対応する種値及び写像関数から検証キーを生成するためのスタンピング手段と、

対応する抽出値を形成するため、写像関数と少なくとも１つのＬＵＴとに従って、複数の周波数領域係数値の一部のそれぞれを写像するための写像手段と、

スタンピング情報のうちの対応する部分と同等の修正された抽出値を有する、それぞれの修正された周波数領域係数値を作るため、スタンピング情報のうちの対応する部分と一致しない抽出値を有する、複数の周波数領域係数値の一部のうちの選択された一部を修正するための手段とを含む、複数の周波数領域係数値によって表されるソース画像を、スタンピング情報を用いてスタンプするためのシステム。

【請求項１８】スタンピング情報が、複数の透かし画像値として形成される透かし画像であることを特徴とする、請求項１７に記載の画像スタンピング・システム。

【請求項１９】複数の周波数領域係数値が、ソース画像の離散コサイン変換（ＤＣＴ）係数値の組であり、複数の周波数領域係数値の一部が、ＤＣＴ係数値の組のうちの直流係数を含むことを特徴とする、請求項１８に記載の画像スタンピング・システム。

【請求項２０】（ａ）少なくとも１つの参照テーブル（ＬＵＴ）に対応する種値及び写像関数から検証キーを生成するステップと、

（ｂ）それぞれの抽出値を形成するため、写像関数及び少なくとも１つのＬＵＴに従って、複数のソース画像値のそれぞれを写像するステップと、

（ｃ）スタンピング情報のうちの対応する部分と同等の変更された抽出値を有する、それぞれの変更されたソース画像値を作るため、スタンピング情報のうちの対応する部分と一致しない抽出値を有する、複数のソース画像値のうちの選択されたソース画像値を変更するステップと、

（ｄ）検証キーによって定義される写像関数及び少なくとも１つのＬＵＴに従って、スタンピング情報を抽出するステップと、

（ｅ）抽出されたスタンピング情報を提供するステップと、

（ｆ）スタンピング情報と抽出されたスタンピング情報とが実質的に同等でない場合に、抽出されたソース画像への変造があることの信号を送出するステップとを含む、複数のソース画像値を有するソース画像を、スタンピング情報を用いてスタンプし、検証する方法。

【請求項２１】（ａ）少なくとも１つの参照テーブル（ＬＵＴ）に対応する種値及び写像関数から検証キーを生成するステップと、

（ｂ）それぞれの抽出値を形成するため、写像関数及びＬＵＴに従って、複数のソース画像値のそれぞれを写像するステップと、

（ｃ）スタンピング情報のうちの対応する部分と同等の修正された抽出値を有する、それぞれの修正されたソース画像値を作るため、スタンピング情報のうちの対応する部分と一致しない抽出値を有する、複数のソース画像値のうちの選択されたソース画像値を修正するステップとを含む、複数のソース画像値を有するソース画像を、スタンピング情報を用いてスタンプする方法。

【請求項２２】複数のソース画像値が、複数のソース画素値であり、スタンピング情報が、複数の透かし画像画素値として形成される透かし画像であり、写像のステップ（ａ）が、さらに、

（ａ）（１）写像関数と、少なくとも１つのＬＵＴと、複数の透かし画像画素値のうちの対応する１つとに従っ

て、スタンプされた画像値のうちの対応する１つを各ソース画素値にセットすることによって、ソース画像に透かし画像を埋め込むステップを含むことを特徴とする、請求項２１に記載の画像スタンピング方法。

【請求項２３】（ａ）１ソース画素と、ソース画像内の、誤差値を含む対応する修正画素とを識別するステップと、

（ｂ）誤差値を含む修正画素の値と、スタンプされたソース画像内の対応する画素の値との間の差として定義される修正画素誤差の量を決定するステップと、

（ｃ）隣接画素値の調整の総量が修正画素誤差と同等になるように、各隣接ソース画素の対応する修正画素が変更され、修正画素値が、スタンピング情報を組み込まれ、隣接する変更される画素値が、所望のスタンピング情報をまだ組み込まれておらず、後続のスタンピング処理で適当なスタンピング情報を組み込むためにさらに修正されることを特徴とする、連続する隣接ソース画素値を、識別されたソース画素から離れるにつれて量が減少する修正画素誤差を用いて調整することによって、修正画素誤差を誤差拡散するステップとを含み、スタンプされた画像が、複数のソース画素値によって定義され、対応する修正画素を形成するためにそれぞれがスタンピング情報を用いてスタンプされるソース画像であることを特徴とする、誤差拡散されたスタンプされた画像を提供する方法。

【請求項２４】（ａ）誤差値を含む１ソース画素値と、スタンプされた画像内の対応する修正画素値とを識別するステップと、

（ｂ）誤差値を含むソース画素値と、対応する修正画素値との間の差として定義される修正画素誤差の量を決定するステップと、

（ｃ）（ｉ）変更される修正画素のそれぞれと（ｉｉ）修正画素とが、修正画素値としてスタンピング情報のうちの実質的に同様の部分を有するように、各隣接ソース画素の対応する修正画素を変更することによって、連続する隣接ソース画素値を、識別されたソース画素から離れるにつれて量が減少する修正画素誤差を用いて調整することによって、修正画素誤差を誤差拡散するステップとを含み、スタンプされた画像が、複数のソース画素値によって定義され、対応する修正画素を形成するためにそれぞれがスタンピング情報を用いてスタンプされるソース画像であることを特徴とする、誤差拡散されたスタンプされた画像を提供する方法。

【請求項２５】（ｄ）誤差拡散されたスタンプされた画像を提供するため、全ソース画素が識別され、誤差拡散されるまで、ステップ（ａ）ないし（ｄ）を反復するステップをさらに含むことを特徴とする、請求項２４に記載の誤差拡散されたスタンプされた画像を提供する方法。

【請求項２６】検証キーに基づいて少なくとも１つの参照テーブル（ＬＵＴ）を生成するための手段と、

少なくとも１つのＬＵＴに対応する写像関数を各ソース画像画素に適用し、写像関数の出力値が複数のスタンピング値のうちの対応する１つに対応するように各ソース画素を変更するための手段とを含む、ソース画像が複数のソース画像画素によって表され、スタンピング情報が複数のスタンピング値であることを特徴とする、ソース画像をスタンピング情報を用いてスタンプするためのシステム。

【請求項２７】検証キーに基づいて少なくとも１つの参照テーブル（ＬＵＴ）を生成するためのテーブル生成手段と、

写像関数の出力値が複数のスタンピング値のうちの対応する１つを抽出するように、少なくとも１つのＬＵＴに対応する写像関数を各スタンプされたソース画像画素に適用するための手段とを含む、スタンプされたソース画像が複数のスタンプされたソース画像画素によって表され、スタンピング情報が複数のスタンピング値であることを特徴とする、スタンピング情報を有するスタンプされたソース画像からスタンピング情報を抽出するためのシステム。

【請求項２８】検証キーが、種値を含み、テーブル生成手段が、種値を使用する乱数ジェネレータを使用することによって、少なくとも１つのＬＵＴの各項目を生成することを特徴とする、請求項２６に記載のスタンプされたソース画像からスタンピング情報を抽出するためのシステム。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】本発明は、ディジタル画像処理の分野に関し、具体的には、マルチメディア・オブジェクト・セキュリティ及び画像の透かし（watermarking）の分野に関する。

【０００２】

【従来の技術及び発明が解決しようとする課題】認証と保護を目的とするメモリに記憶されたディジタル画像の完全性の検証及び保護を含むディジタル画像の検証及び保護の分野は、最近重要になり、広く認識されてきた。

【０００３】この分野では、複数の領域で研究が開始されている。研究領域の１つでは、「信頼に値するディジタル・カメラ」に焦点を合わせ、透かし方式を画像に組み込んで、ディジタル写真が修正されたかどうかを判定する。この透かし処理には、既存の公開鍵暗合技術の使用が含まれる。もう１つの研究領域では、最下位ビット（ＬＳＢ）のビット・プレーン操作による画像への検出不能なディジタル署名のコーディングならびにｍシーケンスを使用する画像データへの透かしの線形加算の適用が研究されている。もう１つの技法では、統計的手法を使用してホスト画像に１ビットのデータを埋め込み（バッチワーク）、ランダムなテクスチャのブロックを使用

して類似したテクスチャの領域を置換して、自己相関測定によって形状が回復された、テクスチャを有する領域の同一の対を作成する（テクスチャ・ブロック・コーディング）ことによって、画像に単純なデータを隠す。また、他の研究には、ランダムに生成されたアドレスの組に「ディジタル封印」と称する２進ビットの長い列を埋め込み、列内の対応するビットと一致するようにランダムに生成されたアドレスの組の画素のＬＳＢを変更する技法が含まれる。

【０００４】上の技法の多くでは、情報が、画像の画素値のＬＳＢに「スタンプ」される。このスタンピング技法は、画像内の視覚的なアーチファクトを発生する可能性がほとんどない。画像を変更する場合、検証によってその修正を判定できるようにするために、ＬＳＢが変更される可能性が最も高い。しかし、そのようなＬＳＢ操作は、悪意のある攻撃に対して安全ではない。すべての画素値のＬＳＢを変更せずに画像の内容を変更するシステムを作成することは、比較的容易であり、実際、画像全体を置換することができるが、すべての画素のＬＳＢが元のソース画像と同一に保たれる限り、検証処理でそのような変更を検出することはできない。多数の既存の技法のもう１つの短所は、検証処理で修正の範囲を判定できないことである。検証処理は、画像が修正されたか否かを判定することはできるが、変更が行われた場所を突き止めることはできない。このような情報は、よりよいセキュリティの尺度のために貴重になる可能性がある。

【０００５】極端な場合、スタンピング処理は、スタンピング処理自体は何もなく、画像にスタンプされる情報（「スタンピング情報」）が画像そのものになる。この場合、検証処理は、やはり何も行わず、抽出されるスタンピング情報は、スタンプされたソース画像自体になる。スタンピング情報と抽出されたスタンピング情報の最終的な比較は、効果的に、スタンピングの時点で既知のソース画像と、検証の時点で既知のソース画像との比較になる。しかし、このスタンピング情報は非常に大きく、情報を記憶するのにかなりのメモリを必要とするので、この比較は非常に効率的ではない。同様に、このスタンピング情報の伝送にはかなりの帯域幅が必要である。この例から、検証処理の望ましい特性の１つが、スタンピング情報の量が非常に少ないことであることがわかる。

【０００６】その一方で、もう１つのスタンピング処理は、画像内の所定の位置から画素を抽出し、その画素と画素位置をスタンピング情報として使用することである。この場合、検証処理では、スタンプされた画像内でこの画素が変更されていないかどうかを判定する。この例では、スタンピング情報の量は少ないが、正しい検証が行われる確実性も低い。というのは、単一の所与の画素に影響しない、多数の異なる画像の変更が存在するからである。したがって、検証処理の第２の望ましい特性は、検証処理によって、高い確度で、スタンピング以降にスタンプされた画像が修正されたかどうかを判定できることである。

【０００７】画像検証のほかに、ディジタル透かしは、著作権保護手段のために提案されてきた。たとえば、米国特許第５５３０７５９号明細書には、画像の無許可使用を防止するため、画像に可視の「透かし」を配置するシステムが開示されている。米国特許第５４８８６６４号明細書には、黒画素と白画素を使用して符号化される印刷された暗号「透かし」を使用して、無許可のアクセス及び修正から視覚情報を保護するための方法及び装置が開示されている。これらは、「可視透かし」の使用の例である。透かし自体は、スタンプされた画像に明示的に表示され、知覚的に容易に目につくようになっている。

【０００８】

【課題を解決するための手段】画像検証処理で、前に画像の内容にスタンプされた時以降に画像の内容が変更されていないことを検証する。画像検証処理では、ソース画像とスタンピング情報を受け取り、スタンピング情報をソース画像に埋め込んで、スタンプされた画像を作る。キーを用いて、スタンプされた画像からスタンピング情報を抽出できるようにする。埋込み処理の後に、画像検証処理で、キーに基づいてスタンプされた画像からスタンピング情報を抽出し、元のスタンピング情報が抽出されたスタンピング情報と一致しない場合には、スタンプされた画像が変造されていると判定する。

【０００９】

【発明の実施の形態】

概要

本発明は、画像内容検証のため画像にスタンプされる「不可視透かし」を介して画像を検証するためのシステム及び方法に関する。透かしは、スタンピング情報からなり、画像スタンピング処理では、透かしの可視の痕跡または画像内の視覚的アーチファクトを全く生成せずに透かし値とソース画像値を組み合わせる。言い換えると、この透かしは知覚的に不可視である。不可視であるから、画像に対する修正で透かしが必ず変更されるが、可視の特徴とは異なり、攻撃者は、画像にスタンプされた変更された不可視のマークを復元できない。したがって、真正性について画像の内容を検証できる。さらに、挿入される透かしは、不可視ではあるが、後程抽出して、所有権情報を示すことができる。

【００１０】本発明のシステムは、前に画像の内容にスタンプされた時以降に画像の内容が変更されていないことをすばやく検証する。このシステムは、スタンプされた画像を作るために定義された写像処理に基づいて、スタンピング情報と称するディジタル情報をソース画像に埋め込むスタンピング処理からなる。写像処理は、復号

「キー」によって識別され、この復号キーを用いると、ユーザは、スタンプされた画像からスタンピング情報を復号でき、したがって、このキーを検証キーと称する。このシステムには、検証キーによって識別された写像処理に基づいて、スタンプされたソース画像からスタンピング情報を抽出する検証処理も含まれる。本発明の１実施例では、スタンピング情報は、透かし画像（watermark image）と称する所定の画像を定義する情報でもある。この透かし画像を用いると、検証処理で抽出されたスタンピング情報を、スタンピング処理で適用されたスタンピング情報と視覚的に比較できるようになる。もう１つの実施例では、この比較を、コンピュータ・アルゴリズムによって実行できる。最後に、スタンピング情報は、スタンプされた画像の正しい所有権を示すために表示できる。

【００１１】本発明は、スタンプされた画像からスタンピング情報を抽出し、画像がスタンプされた時に追加された情報と比較することによって、画像の内容が変更されていないことをすばやく検証する。一致が得られる場合、画像は、スタンプ以降に変更されていないと判定される。本発明には、ソース画像に知覚可能な変化を生じることなく、ソース画像の画像値にスタンピング情報を埋め込むためのシステムも含まれる。したがって、スタンピング情報は、正しい検証キーを用いて簡単かつ安全に検証目的のために抽出でき、検証キーがなければ、スタンピング情報は取得できない。

【００１２】本発明のこの実施例では、ソース画像画素値にスタンピング情報を埋め込む処理によって、誤差が導入され、画素値が変更される。しかし、導入される誤差は、誤差拡散処理によって局所的に分散されるので、画像の視覚的な品質低下を引き起こさない。さらに、スタンピング情報は、ソース画素値と導入される誤差の組み合わせによって埋め込まれ、この情報は、最下位ビット（ＬＳＢ）より上位のさまざまなビット値に隠され、したがって、本発明は、ＬＳＢ操作の短所をこうむらず、攻撃に対してより安全になっている。さらに、本発明は、検証処理で画像変更の領域を判定し、その位置を限定することができる。

【００１３】本発明のスタンピング情報は、任意のディジタル・ビット・ストリームの形とすることができる。好ましい実施例では、商標及びグラフィック記号の形で所有権情報を明瞭に描写する画像を、スタンピング情報として使用する。したがって、本発明では、スタンピング情報も画像であり、これを「透かし画像」と称する。以下の説明では、透かし画像という単語は、スタンピング情報の１形式と同等であり、したがって、これらは交換可能である。

【００１４】ａ）不可視画像スタンピング処理の要約
不可視画像スタンピング処理では、透かし画像Ｗ（Ｉ，Ｊ）をソース画像Ｓ（Ｉ，Ｊ）に埋め込んで、スタンプされたソース画像ＳＳ（Ｉ，Ｊ）を作る（Ｉ及びＪは、たとえばＩ列Ｊ行を表す、行列内の特定の値を示す整数である）。ソース画像内の各画素は、順番に処理される。この処理では、選択された画素に透かし抽出関数ＷＸ（*）を適用し、抽出された透かし値をテストして、埋め込まれる透かしの値と等しいかどうかを判定する。これらが等しい場合、処理は次の画素に進む。等しくない場合には、抽出された透かしの値が埋め込まれる透かしの値と等しくなるまで選択された画素の値が変更され、これを行うのに必要な変更が計算され、その変更と反対の変更が、誤差拡散を使用して、まだ処理されていない画素に伝播される。ソース画像内のすべての画素がスタンプされるまで、この処理を繰り返す。最終的な産物としてのスタンプされた画像と共に、検証キーが作られる。

【００１５】ｂ）画像検証処理の要約
画像検証処理では、透かし画像ＥＷ（Ｉ，Ｊ）が、スタンプされたソース画像ＳＳ（Ｉ，Ｊ）から抽出される。この抽出は、検証キーから透かし抽出関数ＷＸ（*）を計算することによって開始され、その関数をすべての画素ＳＳ（Ｉ，Ｊ）に適用して、透かし画素ＥＷ（Ｉ，Ｊ）が作られる。スタンプされたソース画像のすべての画素が処理されるまで、透かし抽出処理を繰り返す。その結果は、抽出された透かし画像である。この画像は、スタンプされた画像の変更及び相違に関する検査のために、視覚的または数値的に元の透かし画像と比較することができる。

【００１６】ｃ）ＪＰＥＧ圧縮画像に対する画像検証とスタンピング
高解像度のディジタル画像は、かなりの量の記憶空間を必要とする可能性があり、伝送コストが高くなる可能性がある。これらの画像の多くは、ＪＰＥＧ圧縮規格に従う圧縮形式で配布される可能性が高い。埋込みスタンピング情報または透かしを有する画像を直接圧縮すると、圧縮技法に固有の量子化処理で画素値のほとんどが変更されるので、透かしのほとんどが破壊される。本発明の好ましい実施例では、ＪＰＥＧ圧縮データ形式にコード化された画像に対して不可視画像スタンピング処理と画像検証処理を実行するための方式も提供される。スタンピング情報は、画像を完全に伸長することなく、既存のＪＰＥＧ圧縮画像データ・ストリームに直接埋め込むことができる。同様に、画像検証は、ＪＰＥＧ圧縮画像から直接行うことができ、圧縮画像を完全に伸長してから内容を検証する必要はない。圧縮されたＪＰＥＧソース画像に対する画像検証及び不可視スタンピングの処理は、ＪＰＥＧ画像データ・ストリームの直流係数を抽出して直流係数画像を形成し、その直流係数画像に対して画像検証及びスタンピングの処理を適用することによって達成される。その後、スタンプされた直流係数を、標準ＪＰＥＧデータ形式に従って、圧縮データ・ストリー

ムに再符号化する。

【００１７】誤差拡散

好ましい実施例では、誤差を導入し、その後、導入された誤差を滑らかに局所的に拡散することによって、画像をスタンプする。この拡散は、誤差拡散処理の修正及び適合によって達成される。

【００１８】画像の中間調化のための誤差拡散技法は、当技術分野で一般に周知である。１つの技法が、フロイド（Robert Floyd）及びシュタインバーグ（Louis Steinberg）著、「An Adaptive Algorithm for Spatial Gray Scale」、1975 SID International Symposium, Digest of Technical Papers、３６～３７ページに記載されている。誤差拡散の多数の変形が、ウリクニ（R. Ulichney）著、「Digital Halftoning」、MIT Press（米国マサチューセッツ州ケンブリッジ）刊、１９８７年にも記載されている。

【００１９】誤差拡散は、各画素位置での拡散誤差$e_{i,j}$を０にすることから始まる。数学的には、これを$e_{i,j}=0 \forall i, j$と表す。入力画素は、初期画素位置または第１画素位置の選択された画素の処理から順番に処理され、この入力画素に対して、拡散誤差が決定され、残りの画素のそれぞれについて、以下の処理ステップが行われる。

１．入力画素値$ip_{i,j}$とその画素位置での拡散誤差の値の和として、修正画素値$mp_{i,j}$を計算する。数学的には、これを

$$mp_{i,j}=ip_{i,j}+e_{i,j}$$

と表す。

２．出力画素値$op_{i,j}$を、修正画素値に近い可能な出力値q_kのうちの１つとして選択する。数学的には、これを

$$op_{i,j}=Q(mp_{i,j})$$

と表す。ただし、Ｑ（ｘ）は、ｘに近い可能な出力値q_kを１つ選択する関数である。

３．ここで、量子化誤差は

$$d_{i,j}=mp_{i,j}-op_{i,j}$$

によって表される。

４．まだ処理されていない画素位置では、この画素位置での量子化誤差に比例する量だけ拡散誤差を増分する。すなわち、$\Sigma c_{r,s}=\gamma$のもとで

$$e_{i+r,j+s}=e_{i+r,j+s}+c_{r,s}d_{i,j}$$

となる。

【００２０】さまざまな誤差拡散技法の相違の多くは、上のステップ４で使用される拡散係数$c_{r,s}$に関して行われる選択の変形である。たとえば、上に示したフロイド及びシュタインバーグの論文では、この係数が定数であるが、ランダムな可変係数の使用の例が、参照によって本明細書に組み込まれる米国特許第４６５４７２１号明細書に記載されている。

【００２１】画像検証システム

図１は、本発明の実施例に従う使用に適した、不可視画像スタンピング処理と画像配布処理のためのシステムのブロック図である。不可視画像スタンピング処理のフローが、本発明の実施例に従う使用に適した記憶及び配布システムと共に図示されている。ブロック１０３の不可視画像スタンピング処理では、ソース画像１０１とスタンピング情報１００を組み合わせて、スタンプされたソース画像１０４を作る。検証キー１０５も作られる。スタンプされたソース画像１０４は、スタンピング情報１００を埋め込まれてはいるが、知覚的にソース画像１０１とほぼ同一に見えなければならない。言い換えると、スタンピング情報は、元の画像値に直接に不可視の形で隠される。このスタンピング処理は、ディジタル・コンピュータ１０２内で実行される。ブロック１０３に示された不可視スタンピング処理を、以下の節で詳細に説明し、図５、図６及び図７に図示する。

【００２２】スタンプされた画像は、画像アーカイブ１０６に記憶でき、その後、画像サーバ１０８によって取り出すことができる。検証キーは、保護記憶装置１０７に記憶される。画像サーバは、スタンプされた画像を制御し、ローカル・エリア・ネットワークまたはインターネットなどの広域ネットワークとすることができるコンピュータ・ネットワーク１０９を介して、ネットワークに接続された個々のクライアント・コンピュータ・システム１１０へ、要求時にスタンプされた画像を配布する。画像サーバ１０８は、検証キーへのアクセスも制御し、特定の画像の正当な権利を有するクライアントに正しいキーを配布する。スタンピング情報は、正しい検証キーを用いて抽出でき、クライアントのコンピュータ・システム内で表示できる。

【００２３】ａ）画像検証処理

図２は、画像サーバ１０８での画像検証処理の機能レベルのブロック図である。画像サーバ１０８は、画像アーカイブ１０６に記憶された画像データの完全性を検証する。この検証は、まずアーカイブからスタンプされたソース画像１０４を選択し、キーの保護記憶装置１０７からスタンプされた画像に対応する検証キー１０５を取得した後に、この２つの情報を処理のため計算装置に読み取ることによって達成される。検証処理では、スタンプされたソース画像と検証キーをブロック２０１で処理して、画像に埋め込まれたスタンピング情報を抽出する。次に、抽出されたスタンピング情報２０２と、サーバに既知のスタンピング情報１００を、ブロック２０３で比較する。一致が得られる時には、この画像はスタンプされた時以降に変更されていないと判定される。その後、この検証の結果をブロック２０４で使用して、内容の完全性を確認し、内容が変更されていることが検証される場合には適当な処置を行う。サーバの場合、この処置には、可能な不正使用に対するシステム管理の変更ならびに、内容の完全性を調査されている画像のアクセスの制

限を含めることができる。その後、サーバは、ブロック２０５で現画像インデックスを更新し、検証処理のための次の画像の再帰的な形での選択に進む。

【００２４】図３は、クライアント・コンピュータ・システム１１０での画像検証処理の機能レベルのブロック図である。この検証処理は、図２の検証処理に類似しているが、複数の変更を有する（以下では、同一の機能性を示す図２のブロックの符号を、括弧に囲んで示す）。このシステムは、スタンプされたソース画像１０４及び対応する検証キー１０５をサーバから受け取る。次に、スタンプされた画像と検証キーを、ブロック３０１（２０１）で処理して、埋め込まれたスタンピング情報を抽出する。

【００２５】抽出されたスタンピング情報３０２（２０２）は、ブロック３０３で、コンピュータ表示モニタ上に所有権情報を示すために表示することができる。その後、（透かし画像の形で）表示されたスタンピング情報を、ブロック３０４で視覚的な検査に使用して、画像が全体的または部分的に変更されているかどうかを判定できる。画像が変更されていない場合、ブロック３０５で、後程使用するためにスタンピング情報を記憶でき、その結果、最初の抽出の後にブロック３０５で記憶されたスタンピング情報と、現在抽出されているスタンピング情報をブロック３０６（２０３）で比較して、一致するかどうかを調べることによって、受取の後に画像を周期的に検証することが可能になる。この比較でなんらかの相違が見つかる場合、このシステムは、ブロック３０７（２０４）で、悪意のある行為に対する保護のために適当な処置を行うことができる。

【００２６】この検証処理では、ブロック２０１及びブロック３０１で、スタンプされたソース画像と検証キーを処理して、抽出されたスタンピング情報を作る。本明細書では、スタンプされた画像からのスタンピング情報（たとえば透かし画像）の抽出を、「透かし抽出関数」と称する写像関数に関して説明する。しかし、ソース画像にスタンプする処理は、抽出処理に使用される写像関数の逆関数を適用することによって実行される。

【００２７】ブロック２０１及びブロック３０１に示された抽出処理の好ましい実施例の流れ図を、図４に示す。透かし画像ＥＷ（Ｉ，Ｊ）は、スタンプされたソース画像ＳＳ（Ｉ，Ｊ）から抽出される。この処理はステップ４０１から開始され、ステップ４０２で、検証キーから透かし抽出関数ＷＸ（・）を計算し、ステップ４０３で、画素インデックスＩ及びＪを、最初に処理する画素のインデックスに初期設定する。

【００２８】その後、ステップ４０４で、透かし抽出関数を画素ＳＳ（Ｉ，Ｊ）に適用して、その画素での抽出された透かしＥＷ（Ｉ，Ｊ）を、次式によって求める。

ＥＷ（Ｉ，Ｊ）＝ＷＸ（ＳＳ（Ｉ，Ｊ））

【００２９】その後、ステップ４０５で、次に処理する画素に画素インデックスを増分し、ステップ４０６で、画素インデックスを検査する。スタンプされたソース画像内のすべての画素を処理済みの場合、透かし抽出処理はステップ４０７で終了する。スタンプされたソース画像にまだ処理されていない画素がある場合、ステップ４０４で透かし抽出関数を次の画素に適用し、ステップ４０５で画素インデックスをもう一度更新し、ステップ４０６でインデックスをもう一度検査して、スタンプされたソース画像内のすべての画素を処理したかどうかを判定する。

【００３０】ステップ４０２の透かし抽出関数の計算と、ステップ４０４の透かし抽出処理を実施する、本発明の好ましい実施例を、図１０に示す。スタンプされたカラー画像ＳＳ（Ｉ，Ｊ）の場合、画像内のすべての画素（Ｉ，Ｊ）に、それぞれ赤、緑及び青の色の輝度を示す３つの色成分値すなわち、SS_R（Ｉ，Ｊ）、SS_G（Ｉ，Ｊ）及びSS_B（Ｉ，Ｊ）が含まれる。単色のスタンプされた画像の場合、すべての画素（Ｉ，Ｊ）に、１つの色輝度値すなわちＳＳ（Ｉ，Ｊ）だけが含まれることが望ましい。透かし抽出関数ＷＸ（・）は、ステップ４０２で検証キーに基づいて計算される関数であり、形式的には、

ＥＷ（Ｉ，Ｊ）＝ＷＸ（SS_R（Ｉ，Ｊ），SS_G（Ｉ，Ｊ），SS_B（Ｉ，Ｊ））

である。ただし、ＷＸ（・）は、入力値（すなわち画素の輝度値）を２進出力値にすることのできる写像関数とすることができ、この写像関数は、検証キーによって与えられる。

【００３１】好ましい実施例では、１画素の色輝度値のそれぞれに異なる写像関数が適用され、単一の色成分のための写像関数のそれぞれを、色成分写像関数と称する。カラー画像の場合、３つの色成分写像関数の組すなわちF_R（・）、F_G（・）及びF_B（・）が必要であり、単色画像の場合、１つの色成分写像関数すなわちＦ（・）が必要である。言い換えると、Ｆ（・）、F_R（・）、F_G（・）及びF_B（・）のそれぞれが、入力値（画素輝度値）を２進出力値に写像できる。

【００３２】これらの異なる写像関数は、画素値がテーブル項目のインデックスとして働き、テーブル項目によって１ビット値（「０」または「１」）出力が与えられる２進参照テーブル（ＬＵＴ）の形で実施される。この場合、検証キーは、１組の２進参照テーブル（ＬＵＴ）（カラー画像の場合は３つのテーブル、単色画像の場合は１つのテーブル）に関係し、この実施例の場合、１組の２進ＬＵＴを生成するために既知の巡回関数ジェネレータに供給できる数（本明細書では種と称する）とすることができる。

【００３３】この実施例の２進ＬＵＴのそれぞれは、次のように生成される。まず、あるインデックスに対して、テーブル項目は「１」または「０」のいずれかにな

り、これはランダムな形で決定される。たとえば、擬似乱数ジェネレータを使用して、乱数分布を有する任意の数としてテーブル項目を生成できる。当技術分野で既知のとおり、これらの擬似乱数ジェネレータでは、「種」と称する入力数を受け取り、ランダムな出力値を再帰的に生成する巡回関数が使用される。本発明の１実施例では、Ｃプログラミング言語の標準ライブラリの関数サブルーチンｓｒａｎｄ（）及びｒａｎｄ（）を使用する。乱数出力値のそれぞれについて、テーブルの値に、乱数出力値が偶数の場合は「０」、乱数出力値が奇数の場合は「１」をセットする。しかし、当業者には明白なとおり、ランダムなテーブル項目は、複数の方法で生成でき、たとえば、線形フィードバック・シフト・レジスタによって生成される値、乱数表の値、または他の直交関数シーケンスを使用することによって生成できる。

【００３４】ステップ４０４の透かし抽出ステップを、図１０に示す。スタンプされた画像の画素ＳＳ（Ｉ，Ｊ）１００１に対して、色成分のそれぞれの輝度値SS_R（Ｉ，Ｊ）、SS_G（Ｉ，Ｊ）及びSS_B（Ｉ，Ｊ）が、それぞれ赤、緑及び青の対応する２進参照テーブル（ＬＵＴ）すなわちLUT_R、LUT_G及びLUT_B１００２へのインデックスとして働く。各インデックスのテーブル項目を、出力として読み取る。色のそれぞれについて、テーブルから１ビット出力（「０」または「１」）すなわちＶ１、Ｖ２及びＶ３が得られる。その後、これら３つの出力ビットのＸＯＲ（排他的論理和）演算１００３を行って、第（Ｉ，Ｊ）画素の最終的な所望の透かしビット値ＥＷ（Ｉ，Ｊ）１００５を得る。数学的に言えば、排他的論理和演算子を◎によって表すものとして、カラーのスタンプされた画像の場合は

ＥＷ（Ｉ，Ｊ）＝LUT_R（SS_R（Ｉ，Ｊ））◎LUT_G（SS_G（Ｉ，Ｊ））◎LUT_B（SS_B（Ｉ，Ｊ））

である（単色画像の場合はＥＷ（Ｉ，Ｊ）＝ＬＵＴ（ＳＳ（Ｉ，Ｊ））。たとえば、画像の画素SS'が、

SS_R'＝１３４、SS_G'＝２５５、SS_B'＝２５５

の色輝度値を有し、

LUT_R（１３４）＝０、LUT_G（２５５）＝０、LUT_B（２５５）＝１

である場合、SS'の抽出される透かし値EW'は

EW'＝０◎０◎１＝１

になる。ただし、下記の数１を◎で表す。

【数１】

⊕

【００３５】上の例では、ＸＯＲ演算１００３に示されるように透かし抽出関数ＷＸ（*）がＸＯＲ関数の組み合わせであり、写像関数はＬＵＴによって与えられる。多くの他の可能な写像関数及び写像関数の組み合わせを、透かし抽出関数として実施できる。

【００３６】ブロック２０３及びブロック３０６に示された、抽出された透かし値ＥＷ（Ｉ，Ｊ）と透かし画像値Ｗ（Ｉ，Ｊ）の比較は、次のように自動的に達成できる。画素（Ｉ，Ｊ）のそれぞれについて、ＥＷ（Ｉ，Ｊ）とＷ（Ｉ，Ｊ）の間の差の絶対値を記録する。２進透かし値の場合、この値は、２つの値が一致する場合は「０」、２つの値が一致しない場合は「１」になる。すべての画素の差の絶対値を加算し、その和が所定の閾値を超える場合には、スタンプされた画像が変更されていると宣言する。他の自動比較アルゴリズムも実施できる。

【００３７】不可視画像スタンピング処理

不可視画像スタンピング処理の実施例を示す機能レベルのブロック図を、図５に示す。このブロック図は、図１のブロック１０３の機能的構成要素を詳細に示す図である。処理ステップの流れ図を、図６及び図７に示し、その詳細を後の節で示す。

【００３８】一番上のレベルでは、ソース画像５１１（Ｓ（Ｉ，Ｊ））に透かし画像５１０の形でスタンピング情報を埋め込んで、可視のアーチファクトをこうむらずにスタンプされた画像５１３（ＳＳ（Ｉ，Ｊ））を作る処理は、ブロック５０２、５０３、５０５、５０６及び５０９に示された複数の主要な構成要素からなる。

【００３９】スタンプされた画像のトリミングされた部分を検証する能力を組み込むため、本発明の１実施例では、スタンピング情報すなわち透かし画像を、ソース画像の少なくとも１つの部分、潜在的には複数の部分に埋め込む。多くの応用分野では、透かし画像のサイズが、ソース画像のサイズと異なり、はるかに小さいことがしばしばであることに注目されたい。したがって、ブロック５０９で、透かし画像のサイズをソース画像と同一のサイズに変更して、ソース画像のすべての画素に透かし情報を埋め込めるようにしなければならない。本発明の好ましい実施例では、このサイズ変更が、サイズの小さい透かし画像を、水平方向と垂直方向の両方に、ソース画像のサイズまで複写する（複写された透かし画像のうちでソース画像の境界の外側の部分は、トリミングされる）ことによって達成される。サイズを変更された透かし画像は、ソース画像と同一の寸法を有するが、Ｗ（Ｉ，Ｊ）によって表され、このサイズを変更された画像を、本発明の説明の残り全体を通じて透かし画像として扱う。

【００４０】機能ブロック５０２は、正しいキーがなければ抽出処理が保護されるようするための、画素から透かし値を抽出する関数の組と検証キーの生成を示す。

【００４１】ブロック４０２で説明したように、好ましい実施例の１つでは、検証キーを、２進参照テーブル（ＬＵＴ）の組とすることができ、また、必要な２進ＬＵＴの生成に使用できる数の既知の関数ジェネレータと１つの数とすることができる。キーを作るために、乱数ジェネレータ５０１を使用して、数のシーケンスをランダムに生成し、その後、この数のそれぞれを、既知の関

数を介してビット値に写像でき、各ビット値は、検証キー５１２として働く２進ＬＵＴの１項目になる。カラーのソース画像の場合は３つの２進ＬＵＴが必要であり、単色のソース画像の場合は１つの２進ＬＵＴが必要である。もう１つの手法では、乱数ジェネレータ５０１が、検証キー５１２として働く種の数を生成でき、この数を、必要な２進ＬＵＴを再現するのに使用される既知の巡回関数ジェネレータに供給することができる。

【００４２】１組の２進参照テーブルがランダムに生成される際に、テーブル項目に同一の値を有する多数の連続した項目が含まれる可能性がある。この状況は、所望の出力ビット値を得るため、画像スタンピング処理でより大きいレベルへの画素値の調整をもたらし、このため、スタンプされた画像に望ましくないアーチファクトが発生する可能性がある。したがって、本発明のもう１つの実施例では、同一の値を有する連続する項目の数を制限して、画素値（したがって、画素の輝度）の潜在的に大きい調整を克服する。本発明のこの実施例では、連続するテーブル項目値の数が小さくなるように制限するために参照テーブルを処理する。本明細書で説明する実施例の場合、許容可能な連続するテーブル項目の最大個数は、４または５である。

【００４３】２進ＬＵＴは、ブロック５０３に示される、ソース画像５１１すなわちＳ（Ｉ，Ｊ）の透かし値の計算で、２進写像関数として働く。透かし値の計算または抽出は、機能性においては前の節で詳細に説明した図４のステップ４０４と同一である。唯一の相違は、入力が、スタンプされた画素値ではなくソース画像の画素値Ｓ（Ｉ，Ｊ）であることである。カラーのソース画像画素Ｓ（Ｊ，Ｊ）の場合、赤、緑及び青の３つの色成分は、それぞれS_R（Ｉ，Ｊ）、S_G（Ｉ，Ｊ）及びS_B（Ｉ，Ｊ）によって表され、３つのＬＵＴによって提供される３つの写像関数の組は、それぞれ$LUT_R(\cdot)$、$LUT_G(\cdot)$及び$LUT_B(\cdot)$によって表される。単色画像の場合、画素の輝度値はＳ（Ｉ，Ｊ）によって表され、２進ＬＵＴによって提供される写像関数はＬＵＴ（・）によって表される。透かし抽出関数ＷＸ（・）は、写像関数の関数である。画素（Ｉ，Ｊ）のそれぞれについて、抽出される透かし値ＥＷ（Ｉ，Ｊ）は、◎がＸＯＲ演算子であるものとして、カラーのソース画像の場合は

$$EW(I,J)=LUT_R(S_R(I,J))\circledcirc LUT_G(S_G(I,J))\circledcirc LUT_B(S_B(I,J))$$

として計算され、単色画像の場合は、ＥＷ（Ｉ，Ｊ）＝ＬＵＴ（Ｓ（Ｉ，Ｊ））として計算される。

【００４４】その後、ブロック５０４に示された比較処理で、ソース画像画素の計算された透かし値を、サイズを変更された透かし画像Ｗ（Ｉ，Ｊ）の所望の値と比較して、これらの値が一致するかどうかを調べる。

【００４５】ゲートウェイ５０７は、２つの部分を有する画素修正の反復処理の実施を制御する。第１の部分では、ステップ５０５で、最初のテストで透かし値の一致が見つからない場合に、計算された透かし値を、透かし画像の対応する画素の所望の値に強制的に一致させるように、ソース画素画素値のそれぞれを変更する。画素変更の場合、修正によって導入される誤差が、スタンプされた画像に可視のアーチファクトを作ってはならないので、誤差は、画像画素輝度値を徐々に変更するために、所定のゆっくり増加するレベルで導入される。したがって、画素修正の処理は、複数回の反復を要する場合がある。変更された画素の透かし値は、新しい誤差調整反復のそれぞれについてブロック５０８で再計算されて、ブロック５０４で、再計算された値が所望の値と一致するかどうかが検査される。第２の部分では、計算された透かし値が所望の透かし値と一致した後に、反復を停止し、ブロック５０６で、近接する画素の平均色輝度値を一定に保つ形で、画素値変更ステップで導入された誤差を近接する画素に拡散する。

【００４６】好ましい実施例の１つでは、カラー画像の画素の場合に、画素の３つの色の値が変更される。画素値は、ひずみが最小になる形で修正されなければならない。どの色輝度値でも、所与の反復で修正できる。調整のレベルは、＋１または－１から開始され、後続の反復で徐々に増加する。各色成分の調整の現在レベルの記録を使用して、レベル変更の大きさが徐々に増加するようにする。変更が画像の視覚的品質の大きな相違に繋がらないように、調整の最大レベルを制御する。

【００４７】所与の反復で変更される色の選択は、事前に定義されたテーブルを使用するか、選択処理に乱数ジェネレータを組み込むことによって達成できる。乱数ジェネレータ使用の利点は、修正が、大多数の画素について特定の色に集中するのではなく、３色のすべてに分散することである。この効果は、大きい面積が単一の輝度値になる可能性がある合成画像の符号化で特に顕著である。修正する画素の選択に規則正しい方式を用いると、画像内の特定の色成分に変更が集中する可能性があり、したがって、これは望ましくない。レベル調整の符号（＋または－）の選択も、局所的に作られる誤差を滑らかにするために、画素ごとにランダムに行われる。

【００４８】単色画像の場合、画素変更は、カラー画素の変更に使用された方法の単純な適応である。画素値の漸進的な調整は、３つの色成分ではなく１つの輝度成分に限られる。

【００４９】上で説明した画素値変更によって導入される誤差は、ブロック５０６に示されるように拡散され、その結果、平均画像輝度値が保存される。これによって、変更を滑らかにするために誤差が画像上で局所的に拡散されるので、正しい平均色が作られる。

【００５０】誤差拡散は、真の色から制限付きカラー・パレットへの変換の際に、近接する画素の色の変化を滑

19

らかにするのに一般的に使用されている。この処理では、急激な変化の代わりに自然なテクスチャと色の変化を追加することによって、高品質のカラー画像が作られる。誤差拡散は、同様の理由から、画素値の変更によって生じる誤差（色変換処理での量子化誤差と同等）が近接する画素に分散するように、符号化装置で使用される。この方法を用いると、正しい平均色を保持でき、望ましくない局所アーチファクトが除去される。本発明には、本発明の具体的な目的に適合された、前に説明した＊

$$MP(I,J)=S(I,J)+E(I,J) \quad (1)$$

２．出力画素値ＳＳ（Ｉ，Ｊ）は、所望の透かし値と一致するビット出力を与える、変更された修正画素値である。※

$$D(I,J)=MP(I,J)-SS(I,J) \quad (2)$$

４．まだ処理されていない近接する画素位置で、現在位置での出力誤差に比例する量だけ拡散誤差を調整する。★

$$E(I+R,J+S)\leftarrow E(I+R,J+S)+C(R,S)D(I,J) \quad (3)$$

を行う。式（１）及び式（２）を組み合わせて、次式が☆

$$SS(I,J)=MP(I,J)-D(I,J)=S(I,J)+E(I,J)-D(I,J) \quad (4)$$

拡散係数の合計$\Sigma_{R,S}C(R,S)$を１に制限することによって、$\Sigma E(I,J)=\Sigma D(I,J)$が得られる。すべての（Ｉ，Ｊ）に対して（４）を合計すると、次式が得られる。

$$\Sigma SS(I,J)=\Sigma S(I,J) \quad (5)$$

【００５２】これによって、平均輝度値が保存され、正しい平均色が作られることが保証される。３つの色ＲＧ◆

$$C(R,S)=\begin{cases} 0.5, & (R,S)=(1,0),(0,1)\text{の場合} \\ 0, & \text{それ以外} \end{cases} \quad (6)$$

【００５４】単色画像の場合、誤差の拡散は、画像の輝度値に限られ、上で説明した方式から簡単に適合できる。

【００５５】このスタンピング処理が完了した時に、スタンプされた画像５１３が、出力として作られ、その画素値にスタンピング情報が組み込まれている。出力のスタンプされた画像５１３のほかに、検証キー５１２が、後続の画像検証処理の目的のため、この処理で作られる。

【００５６】本発明の画像スタンピング処理の流れ図を、図６に示す。画像スタンピング処理は、ステップ６０１で開始され、ステップ６０２で検証キーを計算し、ステップ６０３で透かし抽出関数すなわちＷＸ（＊）を計算し、ステップ６０４で、処理される画素のインデックスＩ及びＪを初期設定する。その後、ステップ６０５で、位置（Ｉ，Ｊ）の画素を処理する。ステップ６０５は、誤差の導入によって位置（Ｉ，Ｊ）の画素値を修正し、近接する画素に誤差を拡散する反復処理である。このステップを、図７に詳細に示す。ステップ６０８で、

20

＊誤差拡散処理の適合が含まれる。

【００５１】Ｅ（Ｉ，Ｊ）を、画素（Ｉ，Ｊ）の所与の色の拡散誤差と定義する。すべての（Ｉ，Ｊ）について、Ｅ（Ｉ，Ｊ）を０に初期設定する。すべての画素（Ｉ，Ｊ）について、以下の誤差拡散処理ステップを行う。

１．現在の修正画素値ＭＰ（Ｉ，Ｊ）は、ソース画像画素値Ｓ（Ｉ，Ｊ）と、その画素位置での拡散誤差の値の合計である。すなわち、

※３．出力誤差Ｄ（Ｉ，Ｊ）は、現在の修正画素値と出力画素値の間の差として計算される。すなわち、

★したがって、いくつかの（Ｒ，Ｓ）について、

☆得られる。

◆Ｂについて、各色の誤差拡散処理は独立に実行される。言い換えると、同一の定式化を、赤（Ｒ）、緑（Ｇ）及び青（Ｂ）の輝度値に適用できる。誤差も、結合された色の誤差拡散処理の変形を使用して拡散できる。

【００５３】本発明の好ましい実施例では、以下の拡散係数を使用する。

画素インデックス（Ｉ，Ｊ）を次に処理する画素のインデックスに増分し、ステップ６０９で、ソース画像全体を処理したかどうかを判定するテストを行う。ソース画像全体を処理した場合、画像スタンピング処理はステップ６０１で完了する。そうでない場合には、この処理はステップ６０５で次の画素に進む。

【００５７】図７は、ステップ６０５によって説明した処理の流れ図である。ステップ７０１で、ソース画像画素Ｓ（Ｉ，Ｊ）と誤差値Ｅ（Ｉ，Ｊ）を与えられて、修正誤差値ＭＰ（Ｉ，Ｊ）を計算する。ステップ７０２で、透かし抽出関数を使用して、画素ＭＰ（Ｉ，Ｊ）から透かし値を抽出する。その後、ステップ７０３で、値ＥＷをテストして、埋め込まれる透かしの値Ｗ（Ｉ，Ｊ）と等しいかどうかを判定する。

【００５８】値ＥＷとＷ（Ｉ，Ｊ）が一致する場合、ステップ７０４で、出力画素の値ＳＳ（Ｉ，Ｊ）にＳ（Ｉ，Ｊ）と等しい値をセットする。位置（Ｉ，Ｊ）でのこの画素のスタンピング処理は、ステップ７１３で完了する。

【００５９】値ＥＷとＷ（Ｉ，Ｊ）が一致しない場合、この処理は、ステップ７０５に進んで、Δ計算子を初期設定し、ステップ７０６でΔの新しい値を計算し、ステップ７０７で変更された修正画素値ＭＳを作成し、ステップ７０８で、透かし値ＥＭＷをＭＳから抽出する。その後、ステップ７０９で、ＥＭＷをテストして、これがＷ（Ｉ，Ｊ）と等しいかどうかを判定する。

【００６０】値ＥＭＷとＷ（Ｉ，Ｊ）が等しい場合、ステップ７１０で、出力画素の値にＭＳの値をセットし、ステップ７１１で、ＭＰ（Ｉ，Ｊ）とＭＳの間の差を計算する。この差が、ステップ７１２で拡散される誤差である。位置（Ｉ，Ｊ）での画像画素のスタンピング処理は、ステップ７１３で終了する。

【００６１】値ＥＭＷとＷ（Ｉ，Ｊ）が等しくない場合、ステップ７０６に戻ることによって、Δの新しい値を計算し、ステップ７０７でＭＳの新しい値を計算し、ステップ７０８でＥＭＷの新しい値を計算し、ステップ７０９でＥＭＷとＷ（Ｉ，Ｊ）が等しいかどうかを調べる。

【００６２】差Ｄ（Ｉ，Ｊ）は、まだ処理されていない近接画素に拡散される誤差である。位置（Ｉ，Ｊ＋１）及び（Ｉ＋１，Ｊ）の近接画素の拡散誤差Ｅは、

Ｅ（Ｉ，Ｊ＋１）＝Ｃ（０，１）Ｄ（Ｉ，Ｊ）

Ｅ（Ｉ＋１，Ｊ）＝Ｃ（１，０）Ｄ（Ｉ，Ｊ）

である。

【００６３】好ましい実施例では、Ｃ（０，１）とＣ（１，０）は０．５と同等である。以下で、本発明の１実施例の処理の例を説明する。画像画素Ｓ'が、色輝度値

（S'_R，S'_G，S'_B）＝Ｓ'

を有し、これと同一の画素位置での誤差ベクトルＥ'が、色輝度値

（E'_R，E'_G，E'_B）＝Ｅ'

を有すると仮定すると、

S'_R＝１１９，S'_G＝１１，S'_B＝２５０

E'_R＝１，E'_G＝－１，E'_B＝０

かつ

LUT_R（１１９）＝０，LUT_R（１２０）＝０，LUT_R（１２１）＝０

LUT_G（９）　＝０，LUT_G（１０）　＝０，LUT_G（１１）　＝１

LUT_B（２４９）＝０，LUT_B（２５０）＝１，LUT_B（２５１）＝１

の場合には、

ＭＰ'＝Ｓ'＋Ｅ'＝（１２０，１０，２５０）

になり、ＭＰ'で抽出される透かし値は

ＥＷ'＝ＷＸ（ＭＰ'）＝０◎０◎１＝１

になる。所望の透かし値Ｗ'が０の場合、すなわち、ＥＷ'≠Ｗ'の場合、ＭＰ'を修正する。まず、Δを（０，０，０）に初期設定する。次に、新しいデルタを計算する。Δ＝（１，０，０）の場合、修正画素値は、

ＭＳ＝ＭＰ'＋Δ＝（１２１，１０，２５０）

になり、

ＥＭＷ＝ＷＸ（ＭＳ）＝０◎０◎１＝１

になる。ＥＷＸ≠Ｗ'なので、反復が継続する。新しいΔを計算し、たとえばΔ＝（０，１，０）であれば、

ＭＳ＝ＭＰ'＋Δ＝（１２０，１１，２５０）

ＥＭＷ＝ＷＸ（ＭＳ）＝０◎１◎１＝０

になる。今度はＥＭＷとＷ'が一致するので、ＳＳ'にＭＰ'の値をセットし、出力誤差Ｄ'、この例では（０，－１，０）を、近接する画素に拡散する。ＥＭＷがＷ'と一致しない場合、新しいΔを計算し、反復を継続する。

【００６４】ＪＰＥＧ圧縮画像に対する画像スタンピング及び検証

本発明の実施例の装置及び方法は、ＪＰＥＧ圧縮画像のマークにも使用できる。圧縮される画像または入力画像のそれぞれは、画素からなると仮定し、各画素は複数の色成分からなる可能性があると仮定する。たとえば、各画素は、３つの色成分すなわち赤、緑及び青からなるものとすることができる。ｉ行ｊ列の入力画像画素の第Ｃ色成分の値を、Ｐ（Ｃ，ｉ，ｊ）と表す。単一の色成分Ｃについて、画像Ｐ（Ｃ，ｉ，ｊ）を、入力画像の第Ｃカラー・プレーンと称する。ここで、ＪＰＥＧ圧縮を簡単に再検討し、ＪＰＥＧ圧縮画像にスタンピング情報を追加し、その内容を後程検証するための処理ステップを要約する。

【００６５】ＪＰＥＧの基礎の形式では、画像のカラー・プレーンのそれぞれが、８行８列に配置される画素の、重なり合わないブロックに分解される。ＭがＭ行目のブロックを表し、ＮがＮ列目のブロックを表し、ｍがブロック内のｍ行目の画素を表し、ｎがブロック内のｎ列目の画素を表し、ｂ（Ｃ，Ｍ，Ｎ，ｍ，ｎ）が第Ｃカラー・プレーンの（Ｍ，Ｎ）番目のブロックの（ｍ，ｎ）番目の画素であるとすると、０＜ｍ＜９、０＜ｎ＜９のｍ、ｎについて、

ｂ（Ｃ，Ｍ，Ｎ，ｍ，ｎ）＝Ｐ（Ｃ，８'Ｍ＋ｍ，８'Ｎ＋ｎ）

である。

【００６６】次に、ＪＰＥＧでは、離散コサイン変換を使用して、ブロック内の６４個の値を変換する。すなわち、８×８のブロックに配置された６４個の変換係数を作る。離散コサイン変換の後に、ＪＰＥＧ圧縮では、係数のそれぞれをスカラで除算し、その結果を整数値に量子化する処理ステップが必要である。その後のＪＰＥＧ圧縮ステップでは、量子化された変換係数を、ハフマン・コーディングを使用してコード化する。

【００６７】ブロックｂ（Ｃ，Ｍ，Ｎ，ｍ，ｎ）の変換によって作られる離散コサイン変換係数のブロックをＢ（Ｃ，Ｍ，Ｎ，ｐ，ｑ）と表すと、そのブロックの直流係数Ｂ（Ｃ，Ｍ，Ｎ，０，０）は、第Ｃカラー・プレー

ンの（Ｍ，Ｎ）番目のブロックの６４個の値の平均である。この直流係数から、直流係数画像ＢＤ（Ｃ，Ｍ，Ｎ）を次のように定義する。

ＢＤ（Ｃ，Ｍ，Ｎ）＝Ｂ（Ｃ，Ｍ，Ｎ，０，０）

【００６８】直流係数画像は、各次元で入力画像のサイズの１／８である。サンプルの画像を図１１に示し、その緑カラー・プレーンから作られた直流係数画像を図１２に示す。

【００６９】本発明の方法を用いてＪＰＥＧ画像にマークを付けるため、本発明の実施例では、ＪＰＥＧによる直流係数値の計算の後に、スタンピング情報を挿入し、処理を実行する。この処理では、この方法で非圧縮画像にマークするのと正確に同一の形で１つまたは複数の直流係数画像にマークする。このマーキングの後に、ＪＰＥＧ処理は、通常通りコサイン変換係数の量子化とエントロピ・コーディングに進む。

【００７０】１実施例によれば、同一の位置（Ｍ，Ｎ）での複数の直流係数画像の複数の直流係数値を、多色直流係数画像の単一画素に関連する複数の色値として扱う。この場合、マーキングは、同一の数の色を有する非圧縮多色画像のマーキングと同一である。他の実施例では、他のカラー・プレーンの直流係数画像と独立に、１つまたは複数の直流係数画像にマークを付ける。

【００７１】図８は、ＪＰＥＧ圧縮形式のソース画像に対して不可視画像スタンピング処理を達成するための好ましい実施例を示す、機能レベルのブロック図である。ソース画像８０１について、この画像に、ＪＰＥＧ圧縮８０２をかけ、ＪＰＥＧ圧縮画像８０３と称するＪＰＥＧ標準規格に適合するデータ・ビット・ストリームに符号化することができる。ＪＰＥＧ圧縮画像８０３から、ブロック８０４で、そのビット・ストリームから直接に直流係数を取得する。直流係数は、ＪＰＥＧ標準規格では交流係数とは異なる形で別々にコード化される。直流係数画像８０５は、抽出された直流係数を組み合わせることによって形成される。直流画像は、元の画像のアイコン版であり、ＪＰＥＧ圧縮で８×８ブロックを使用する場合には、各次元で１／８のサイズである。その後、この直流画像を、直流係数画像と同一サイズのスタンピング透かし画像８０６と共に、前述の画像スタンピング処理８０７（１０３）への入力ソース画像として扱う。検証キー８０９も作られる。ステップ８０８で、スタンピング処理からスタンプされた直流係数画像が作られ、直流画像内の画素値は、透かし情報を埋め込むために変更される。これは、直流係数画像８０５の直流係数が、それ相応に変更されることを意味する。修正された直流係数は、ブロック８１０で再コード化され、ＪＰＥＧデータ・ストリームに統合されて、標準ＪＰＥＧ圧縮画像８１１が作られる。

【００７２】図９は、ＪＰＥＧ圧縮形式のスタンプされた画像に対する画像検証処理を示す、機能レベルのブロック図である。入力画像は、直流係数に透かし情報を埋め込まれた標準ＪＰＥＧ圧縮画像９０１である。このスタンプされた画像のデータ・ストリームは、ＪＰＥＧ標準規格に適合するので、図９のブロック９０３に類似するブロック９０２で、直流係数をデータ・ストリームから直接に抽出して、スタンピング情報を含む直流係数画像９０３を作る。このスタンプされた直流係数画像は、対応する検証キー９０５と共に、ブロック９０４の画像検証処理への入力として使用される。

【００７３】上の方式について、クロミナンス値のサブサンプリングがあるＪＰＥＧ方式の変形では、透かし画像の埋込みと検証に使用するための透かし画像の後続の抽出が、輝度画素値だけに限られる。

【００７４】上の実施例の主要な望ましい特徴の１つが、ＪＰＥＧ圧縮画像を伸長してから内容を検証する必要なしに、画像の検証及びスタンピングをＪＰＥＧ圧縮画像に対する処理から直接に達成できることである。もう１つの特徴は、ＪＰＥＧ圧縮の量子化動作でほとんどの画素値が変更される可能性が高く、したがって、埋め込まれた透かし情報の大部分が失われるので、情報をＪＰＥＧデータ・ストリームに直接埋め込むことによって、透かし情報がＪＰＥＧ圧縮処理で失われなくなることである。

【００７５】本発明の好ましい実施例を図示し、説明してきたが、これらの実施例は、例としてのみ提供されることを理解されたい。当業者であれば、本発明の趣旨から逸脱することなく、多数の変形、変更及び置換を思い浮かべるであろう。したがって、請求の範囲は、そのような変形のすべてを、本発明の趣旨及び範囲に含まれるものとして含むことが意図されている。

【００７６】まとめとして、本発明の構成に関して以下の事項を開示する。

【００７７】（１）（ａ）少なくとも１つの参照テーブル（ＬＵＴ）に対応する種値及び写像関数から検証キーを形成することと、（ｂ）それぞれの抽出値を形成するため、写像関数と少なくとも１つのＬＵＴとに従って、複数のソース画像値のそれぞれを写像することと、

（ｃ）スタンピング情報のうちの対応する部分と同等の変更された抽出値を有する、それぞれの変更されたソース画像値を作るため、スタンピング情報のうちの対応する部分と一致しない抽出値を有する、複数のソース画像値のうちの選択されたソース画像値を変更することとによって、スタンプされた画像を作るための画像スタンピング手段と、スタンプされた画像を受け取り、写像関数に従ってスタンピング情報を抽出するために検証キーを使用し、抽出されたスタンピング情報を提供するための、画像抽出手段と、スタンピング情報と抽出されたスタンピング情報とが実質的に同等でない場合に、スタンプされた画像への変造があることの信号を送出するための信号送出手段とを含む、複数のソース画像値を有する

ソース画像を、スタンピング情報を用いてスタンプし、検証するための装置。

（２）検証キーが、種値を含み、画像スタンピング手段が、さらに、乱数ジェネレータ及び種値を使用して少なくとも１つのＬＵＴの各項目を生成するためのテーブル生成手段を含むことを特徴とする、上記（１）に記載の画像検証装置。

（３）スタンピング情報が、透かし画像であることを特徴とする、上記（１）に記載の画像検証装置。

（４）検証キーとスタンプされた画像を組み合わせるための組み合わせ手段をさらに含み、画像抽出手段が、スタンプされた画像から検証キーを抽出する手段をさらに含むことを特徴とする、上記（１）に記載の画像検証装置。

（５）ソース画像が、複数のソース画素値から形成され、透かし画像が、複数の透かし画像画素値から形成されることを特徴とする、上記（３）に記載の画像検証装置。

（６）画像スタンピング手段が、さらに、複数のソース画素値のそれぞれをアドレッシングする順序を決定するための手段と誤差拡散手段とを含み、誤差拡散手段が、１ソース画素と、ソース画像内の、誤差値を含む対応する修正画素とを識別するための手段と、誤差値を含む修正画素の値と、スタンプされたソース画像内の対応する画素の値との間の差として定義される修正画素誤差の量を決定するための手段と、隣接画素値の調整の総量が修正画素誤差と同等になるように、連続する隣接ソース画素値を、識別されたソース画素から離れるにつれて量が減少する修正画素誤差を用いて調整する手段を提供することによって、修正画素誤差を誤差拡散するための手段とを含むことを特徴とする、上記（５）に記載の画像検証装置。

（７）少なくとも１つの参照テーブル（ＬＵＴ）に対応する種値及び写像関数から検証キーを形成するための形成手段と、それぞれの抽出値を形成するため、写像関数及び少なくとも１つのＬＵＴに従って、複数のソース画像値のそれぞれを写像するための写像手段と、スタンピング情報のうちの対応する部分と同等の変更された抽出値を有する、それぞれの変更されたソース画像値を作るため、スタンピング情報のうちの対応する部分と一致しない抽出値を有する、複数のソース画像値のうちの選択されたソース画像値を変更するための画素変更手段とを含む、複数のソース画像値を有するソース画像を、スタンピング情報を用いてスタンプするためのシステム。

（８）複数のソース画像値が、複数のソース画素値であり、スタンピング情報が、複数の透かし画像画素値から形成される透かし画像であることを特徴とする、上記（７）に記載の画像スタンピング・システム。

（９）信号送出手段が、さらに、スタンプされる情報及び抽出されたスタンピング情報を記憶するための手段と、比較情報を作るため、スタンプされる情報と抽出されたスタンプされた情報を比較するための手段と、抽出されたソース画像に対する変造を比較情報から検出し、抽出されたソース画像のうち、変造に対応する少なくとも１つの領域を示すための手段とを含む、上記（１）に記載の画像検証装置。

（１０）スタンプされた画像が、第１メモリに記憶された複数のスタンプされた画像のうちの１つであり、複数のスタンプされた画像のそれぞれが、対応する検証キーを有し、検証キーのそれぞれが、第２メモリに記憶された複数のキーのうちの１つであることを特徴とする、上記（１）に記載の画像検証装置。

（１１）第１メモリ内の複数のスタンプされた画像からスタンプされた画像を選択するための手段と、第２メモリに記憶された複数のキーから、スタンプされた画像に対応する検証キーを取り出すための手段とをさらに含む、上記（１０）に記載の画像検証装置。

（１２）さらに、第１メモリ内の複数のスタンプされた画像のそれぞれの完全性を検証するための画像完全性検証手段を含み、画像完全性検証手段が、（１）複数のスタンプされた画像のうちのそれぞれ及び対応する検証キーを選択することと、（２）対応する検証キーに基づいてスタンプされた画像からスタンピング情報を抽出することと、（３）検証キーを用いてスタンピング情報を検証することによってスタンプされた画像の完全性を検証することとを反復することを特徴とする、上記（１１）に記載の画像検証装置。

（１３）（ａ）少なくとも１つの参照テーブル（ＬＵＴ）に対応する種値及び写像関数から検証キーを形成することと、（ｂ）それぞれの抽出値を形成するため、写像関数と少なくとも１つのＬＵＴとに従って、複数の周波数領域係数値のそれぞれを写像することと、（ｃ）スタンピング情報のうちの対応する部分と同等の変更された抽出値を有する、それぞれの変更された周波数領域係数値を作るため、スタンピング情報のうちの対応する部分と一致しない抽出値を有する、周波数領域係数値のうちの選択された周波数領域係数値を変更することとによって、スタンプされた画像を作るための画像スタンピング手段と、スタンプされた画像を受け取り、写像関数及びＬＵＴに従って複数の周波数領域係数値の一部からスタンピング情報を抽出するために検証キーを使用し、抽出されたスタンピング情報を提供するための、画像抽出手段と、スタンピング情報と抽出されたスタンピング情報とが実質的に同等でない場合に、抽出されたソース画像への変造があることの信号を送出するための信号送出手段とを含む、複数の周波数領域係数値によって表されるソース画像を、スタンピング情報を用いてスタンプし、検証するためのシステム。

（１４）複数の周波数領域係数値が、複数のソース画像画素値によって表されるソース画像に対する周波数領域

変換演算によって形成されることを特徴とする、上記（１３）に記載の画像検証システム。

（１５）ソース画像画素値に対する変換演算が、離散コサイン変換（ＤＣＴ）であり、複数の周波数領域係数値の一部が、ＤＣＴ変換された複数のソース画像画素値の直流係数を含むことを特徴とする、上記（１４）に記載の画像検証システム。

（１６）スタンピング情報が、透かし画像であることを特徴とする、上記（１３）に記載の画像検証システム。

（１７）少なくとも１つの参照テーブル（ＬＵＴ）に対応する種値及び写像関数から検証キーを生成するためのスタンピング手段と、対応する抽出値を形成するため、写像関数と少なくとも１つのＬＵＴとに従って、複数の周波数領域係数値の一部のそれぞれを写像するための写像手段と、スタンピング情報のうちの対応する部分と同等の修正された抽出値を有する、それぞれの修正された周波数領域係数値を作るため、スタンピング情報のうちの対応する部分と一致しない抽出値を有する、複数の周波数領域係数値の一部のうちの選択された一部を修正するための手段とを含む、複数の周波数領域係数値によって表されるソース画像を、スタンピング情報を用いてスタンプするためのシステム。

（１８）スタンピング情報が、複数の透かし画像値として形成される透かし画像であることを特徴とする、上記（１７）に記載の画像スタンピング・システム。

（１９）複数の周波数領域係数値が、ソース画像の離散コサイン変換（ＤＣＴ）係数値の組であり、複数の周波数領域係数値の一部が、ＤＣＴ係数値の組のうちの直流係数を含むことを特徴とする、上記（１８）に記載の画像スタンピング・システム。

（２０）（ａ）少なくとも１つの参照テーブル（ＬＵＴ）に対応する種値及び写像関数から検証キーを生成するステップと、（ｂ）それぞれの抽出値を形成するため、写像関数及び少なくとも１つのＬＵＴに従って、複数のソース画像値のそれぞれを写像するステップと、（ｃ）スタンピング情報のうちの対応する部分と同等の変更された抽出値を有する、それぞれの変更されたソース画像値を作るため、スタンピング情報のうちの対応する部分と一致しない抽出値を有する、複数のソース画像値のうちの選択されたソース画像値を変更するステップと、（ｄ）検証キーによって定義される写像関数及び少なくとも１つのＬＵＴに従って、スタンピング情報を抽出するステップと、（ｅ）抽出されたスタンピング情報を提供するステップと、（ｆ）スタンピング情報と抽出されたスタンピング情報とが実質的に同等でない場合に、抽出されたソース画像への変造があることの信号を送出するステップとを含む、複数のソース画像値を有するソース画像を、スタンピング情報を用いてスタンプし、検証する方法。

（２１）（ａ）少なくとも１つの参照テーブル（ＬＵＴ）に対応する種値及び写像関数から検証キーを生成するステップと、（ｂ）それぞれの抽出値を形成するため、写像関数及びＬＵＴに従って、複数のソース画像値のそれぞれを写像するステップと、（ｃ）スタンピング情報のうちの対応する部分と同等の修正された抽出値を有する、それぞれの修正されたソース画像値を作るため、スタンピング情報のうちの対応する部分と一致しない抽出値を有する、複数のソース画像値のうちの選択されたソース画像値を修正するステップとを含む、複数のソース画像値を有するソース画像を、スタンピング情報を用いてスタンプする方法。

（２２）複数のソース画像値が、複数のソース画素値であり、スタンピング情報が、複数の透かし画像画素値として形成される透かし画像であり、写像のステップ（ａ）が、さらに、

（ａ）（１）写像関数と、少なくとも１つのＬＵＴと、複数の透かし画像画素値のうちの対応する１つとに従って、スタンプされた画像値のうちの対応する１つを各ソース画素値にセットすることによって、ソース画像に透かし画像を埋め込むステップを含むことを特徴とする、上記（２１）に記載の画像スタンピング方法。

（２３）（ａ）１ソース画素と、ソース画像内の、誤差値を含む対応する修正画素とを識別するステップと、（ｂ）誤差値を含む修正画素の値と、スタンプされたソース画像内の対応する画素の値との間の差として定義される修正画素誤差の量を決定するステップと、（ｃ）隣接画素値の調整の総量が修正画素誤差と同等になるように、各隣接ソース画素の対応する修正画素が変更され、修正画素値が、スタンピング情報を組み込まれ、隣接する変更される画素値が、所望のスタンピング情報をまだ組み込まれておらず、後続のスタンピング処理で適当なスタンピング情報を組み込むためにさらに修正されることを特徴とする、連続する隣接ソース画素値を、識別されたソース画素から離れるにつれて量が減少する修正画素誤差を用いて調整することによって、修正画素誤差を誤差拡散するステップとを含み、スタンプされた画像が、複数のソース画素値によって定義され、対応する修正画素を形成するためにそれぞれがスタンピング情報を用いてスタンプされるソース画像であることを特徴とする、誤差拡散されたスタンプされた画像を提供する方法。

（２４）（ａ）誤差値を含む１ソース画素値と、スタンプされた画像内の対応する修正画素値とを識別するステップと、

（ｂ）誤差値を含むソース画素値と、対応する修正画素値との間の差として定義される修正画素誤差の量を決定するステップと、

（ｃ）（ｉ）変更される修正画素のそれぞれと（ｉｉ）修正画素とが、修正画素値としてスタンピング情報のうちの実質的に同様の部分を有するように、各隣接ソース

画素の対応する修正画素を変更することによって、連続する隣接ソース画素値を、識別されたソース画素から離れるにつれて量が減少する修正画素誤差を用いて調整することによって、修正画素誤差を誤差拡散するステップとを含み、スタンプされた画像が、複数のソース画素値によって定義され、対応する修正画素を形成するためにそれぞれがスタンピング情報を用いてスタンプされるソース画像であることを特徴とする、誤差拡散されたスタンプされた画像を提供する方法。

（２５）（ｄ）誤差拡散されたスタンプされた画像を提供するため、全ソース画素が識別され、誤差拡散されるまで、ステップ（ａ）ないし（ｄ）を反復するステップをさらに含むことを特徴とする、上記（２４）に記載の誤差拡散されたスタンプされた画像を提供する方法。

（２６）検証キーに基づいて少なくとも１つの参照テーブル（ＬＵＴ）を生成するための手段と、少なくとも１つのＬＵＴに対応する写像関数を各ソース画像画素に適用し、写像関数の出力値が複数のスタンピング値のうちの対応する１つに対応するように各ソース画素を変更するための手段とを含む、ソース画像が複数のソース画像画素によって表され、スタンピング情報が複数のスタンピング値であることを特徴とする、ソース画像をスタンピング情報を用いてスタンプするためのシステム。

（２７）検証キーに基づいて少なくとも１つの参照テーブル（ＬＵＴ）を生成するためのテーブル生成手段と、写像関数の出力値が複数のスタンピング値のうちの対応する１つを抽出するように、少なくとも１つのＬＵＴに対応する写像関数を各スタンプされたソース画像画素に適用するための手段とを含む、スタンプされたソース画像が複数のスタンプされたソース画像画素によって表され、スタンピング情報が複数のスタンピング値であることを特徴とする、スタンピング情報を有するスタンプされたソース画像からスタンピング情報を抽出するためのシステム。

（２８）検証キーが、種値を含み、テーブル生成手段が、種値を使用する乱数ジェネレータを使用することによって、少なくとも１つのＬＵＴの各項目を生成することを特徴とする、上記（２６）に記載のスタンプされたソース画像からスタンピング情報を抽出するためのシステム。

【図面の簡単な説明】

【図１】本発明の実施例に従う使用に適した、不可視画像スタンピング処理と、画像及び検証キーの記憶配布処理の構成要素を含むシステムのブロック図である。

【図２】スタンプされた画像及び検証キーのアクセス及び配布を制御するサーバ内の、画像検証処理システムの実施例の機能ブロックを示すブロック図である。

【図３】スタンプされた画像及び検証キーを受け取るクライアント・コンピュータ・システム内の、画像検証処理システムの実施例の機能ブロックを示すブロック図である。

【図４】画像検証処理で、スタンプされた画像から埋め込まれたスタンピング情報を抽出する方法を示す流れ図である。

【図５】不可視画像スタンピング処理の実施例を示す、機能レベルのブロック図である。

【図６】画像の画素に対する不可視画像スタンピング処理の詳細を示す流れ図である。

【図７】画素値の修正と誤差拡散の処理の方法の詳細を示す流れ図である。

【図８】ＪＰＥＧ圧縮形式のソース画像に対する不可視画像スタンピング処理を示す、機能レベルのブロック図である。

【図９】ＪＰＥＧ圧縮形式のスタンプされた画像に対する画像検証処理を示す、機能レベルのブロック図である。

【図１０】入力画像画素からの透かし値の抽出を示すブロック図である。

【図１１】サンプル画像の例を示す図である。

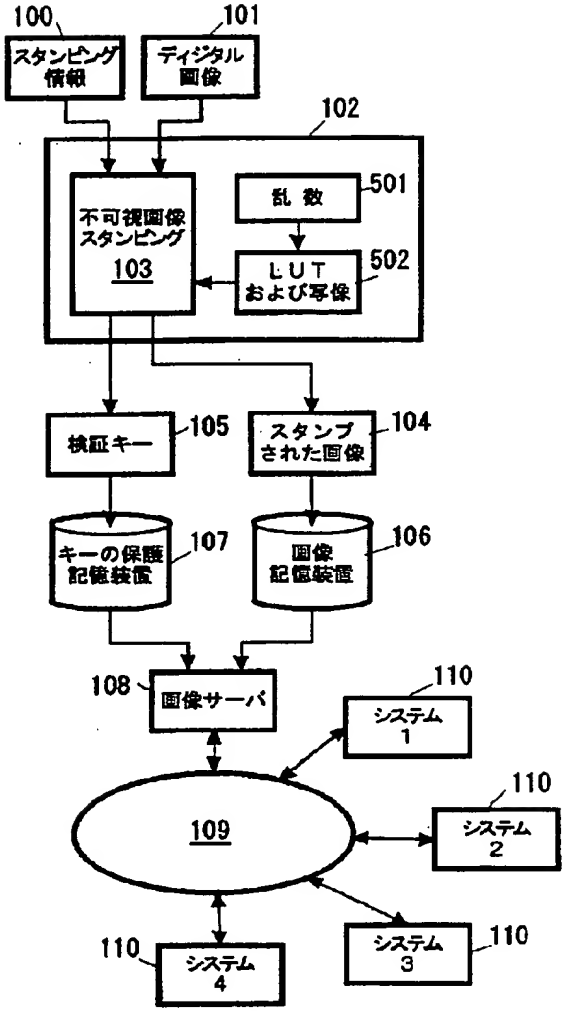
【図１２】図１１に示された画像の直流係数画像の例である。

【符号の説明】

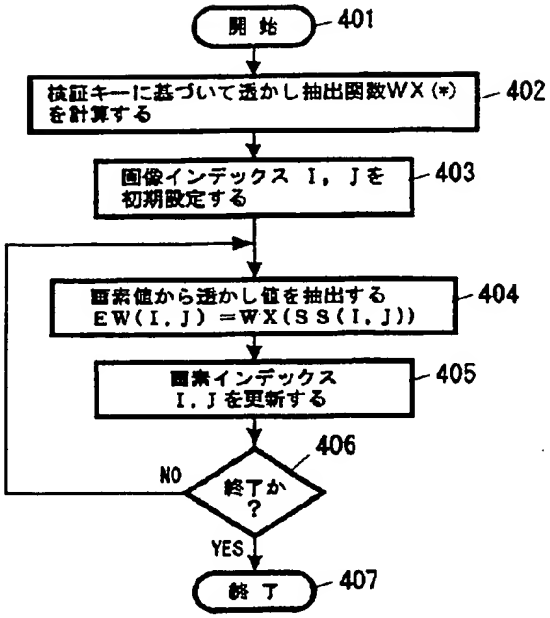
１００　スタンピング情報
１０１　ソース画像
１０２　ディジタル・コンピュータ
１０４　スタンプされたソース画像
１０５　検証キー
１０６　画像アーカイブ
１０７　保護記憶装置
１０８　画像サーバ
１０９　コンピュータ・ネットワーク
１１０　クライアント・コンピュータ・システム
５０１　乱数ジェネレータ

【図１２】

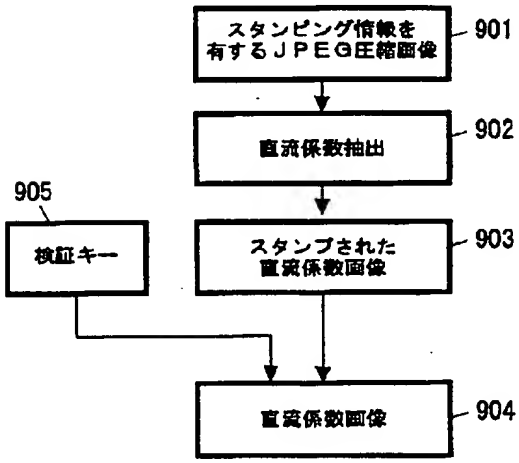
【図1】


100
スタンピング情報
101
ディジタル画像
102
不可視画像スタンピング
103
乱数
501
LUTおよび写像
502
検証キー
105
スタンプされた画像
104
キーの保護記憶装置
107
画像記憶装置
106
108
画像サーバ
110
システム1
109
システム2
システム3
システム4

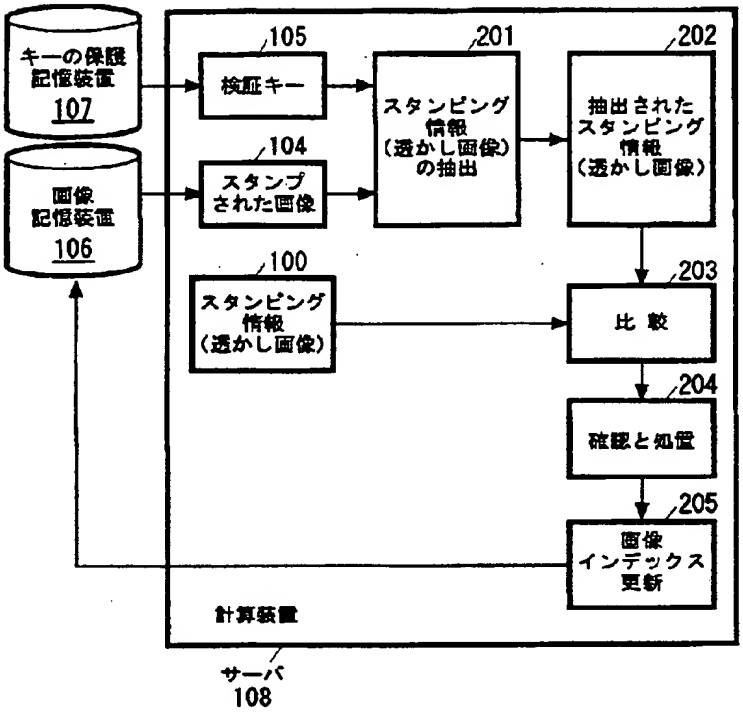
【図4】


開始
401
検証キーに基づいて透かし抽出関数WX(*)を計算する
402
画像インデックスI，Jを初期設定する
403
画素値から透かし値を抽出する EW(I,J)=WX(SS(I,J))
404
画素インデックスI，Jを更新する
405
406
NO
終了か？
YES
終了
407

【図9】


スタンピング情報を有するJPEG圧縮画像
901
直流係数抽出
902
905
検証キー
スタンプされた直流係数画像
903
直流係数画像
904

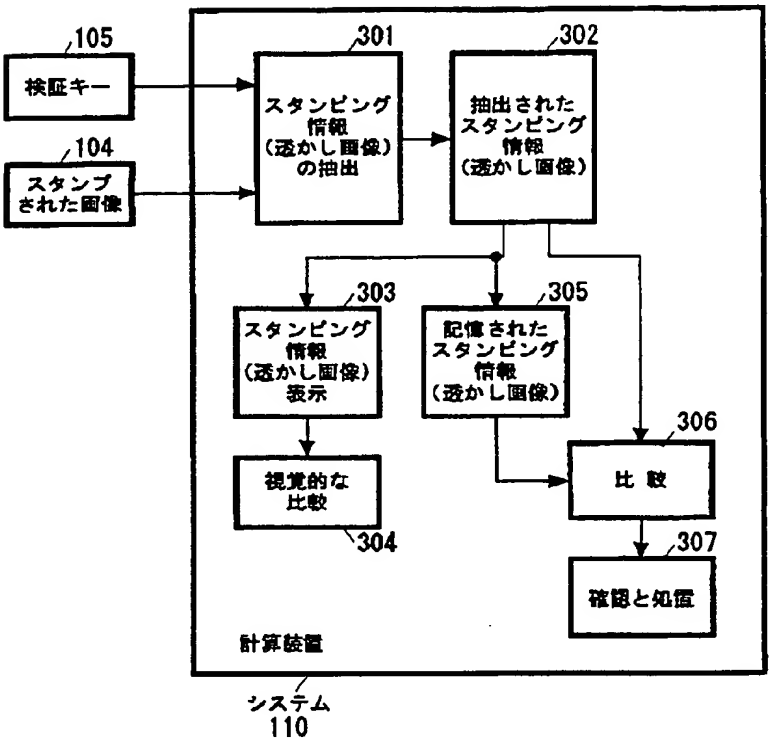
【図２】


キーの保管
記憶装置
107
画像
記憶装置
106
105
検証キー
104
スタンプ
された画像
201
スタンピング
情報
（透かし画像）
の抽出
202
抽出された
スタンピング
情報
（透かし画像）
100
スタンピング
情報
（透かし画像）
203
比 較
204
確認と処置
205
画像
インデックス
更新
計算装置
サーバ
108

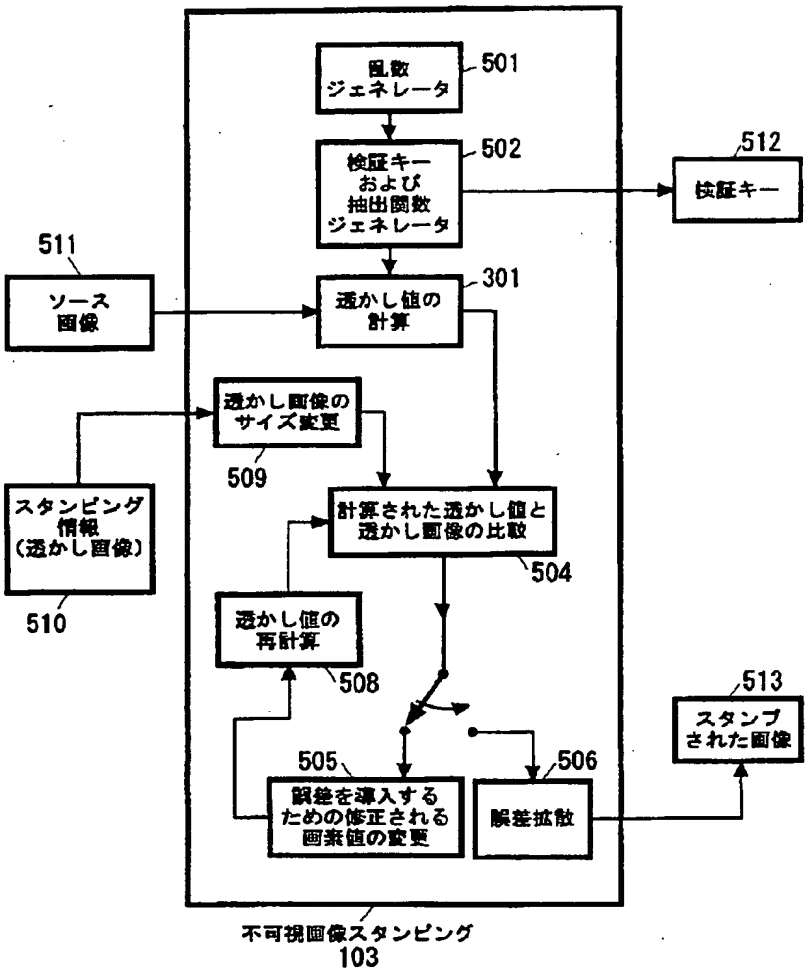
【図１１】


My hard drive's maxed out.

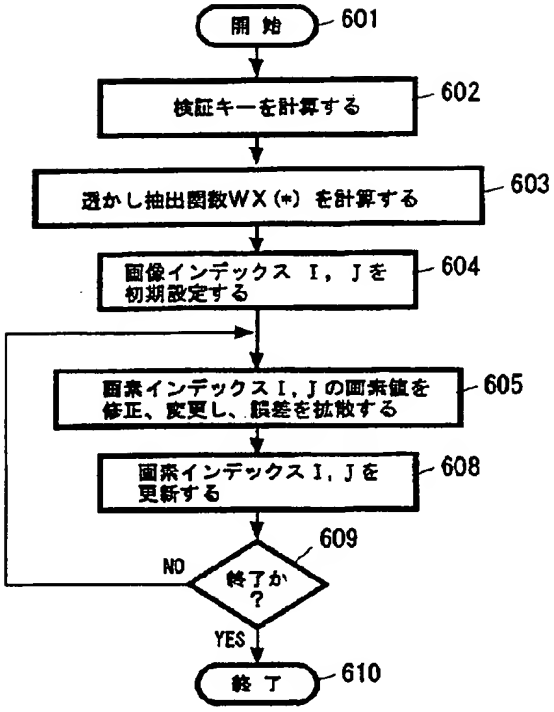
【図３】


105
検証キー
104
スタンプされた画像
301
スタンピング情報（透かし画像）の抽出
302
抽出されたスタンピング情報（透かし画像）
303
スタンピング情報（透かし画像）表示
305
記憶されたスタンピング情報（透かし画像）
306
比 較
視覚的な比較
304
307
確認と処置
計算装置
システム
110

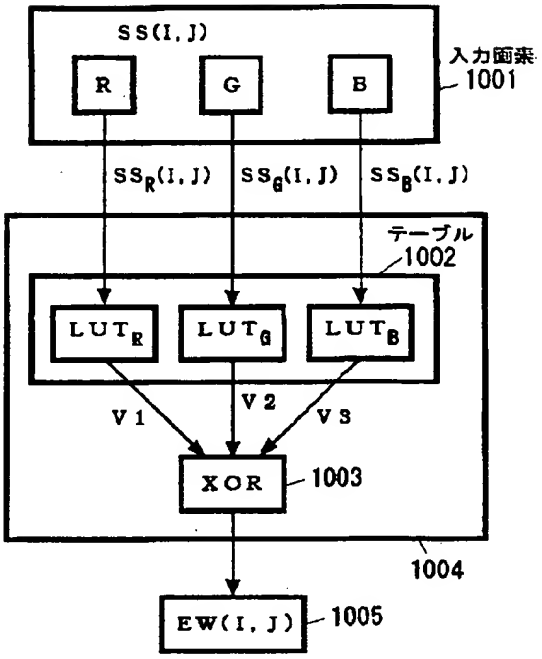
【図５】


乱数
ジェネレータ
501
検証キー
および
抽出関数
ジェネレータ
502
512
検証キー
511
ソース
画像
透かし値の
計算
301
透かし画像の
サイズ変更
509
スタンピング
情報
(透かし画像)
510
計算された透かし値と
透かし画像の比較
504
透かし値の
再計算
508
513
スタンプ
された画像
505
506
誤差を導入する
ための修正される
画素値の変更
誤差拡散
不可視画像スタンピング
103

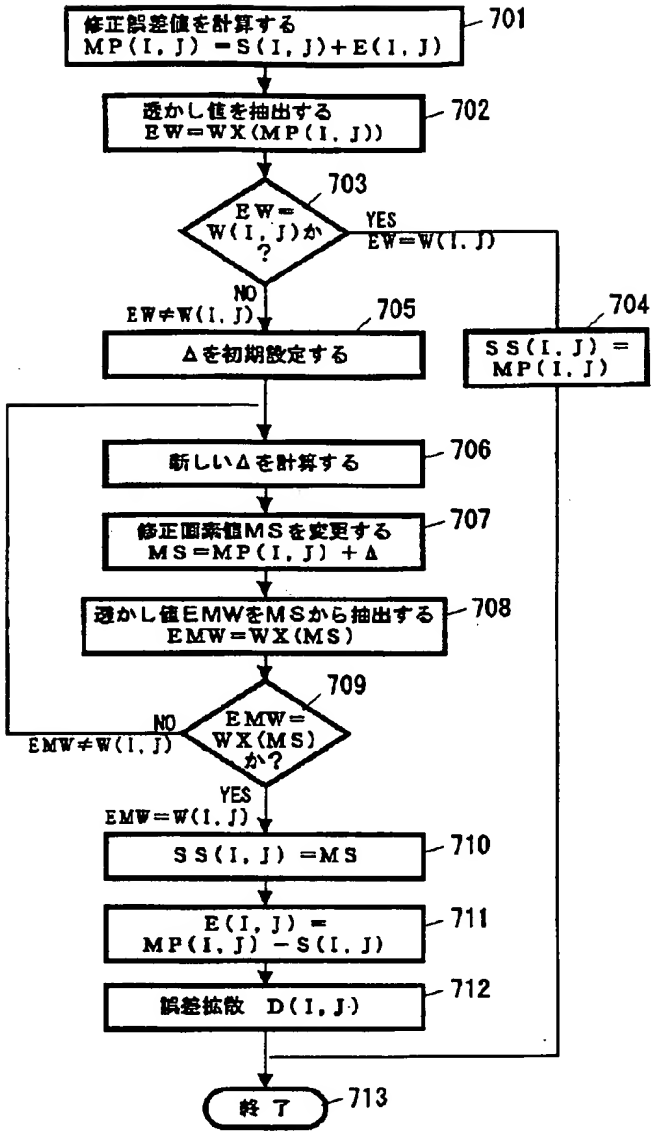
【図６】


開 始
601
検証キーを計算する
602
透かし抽出関数WX(*) を計算する
603
画像インデックス I，Jを
初期設定する
604
画素インデックスI，Jの画素値を
修正、変更し、誤差を拡散する
605
画素インデックスI，Jを
更新する
608
終了か
？
609
NO
YES
終 了
610

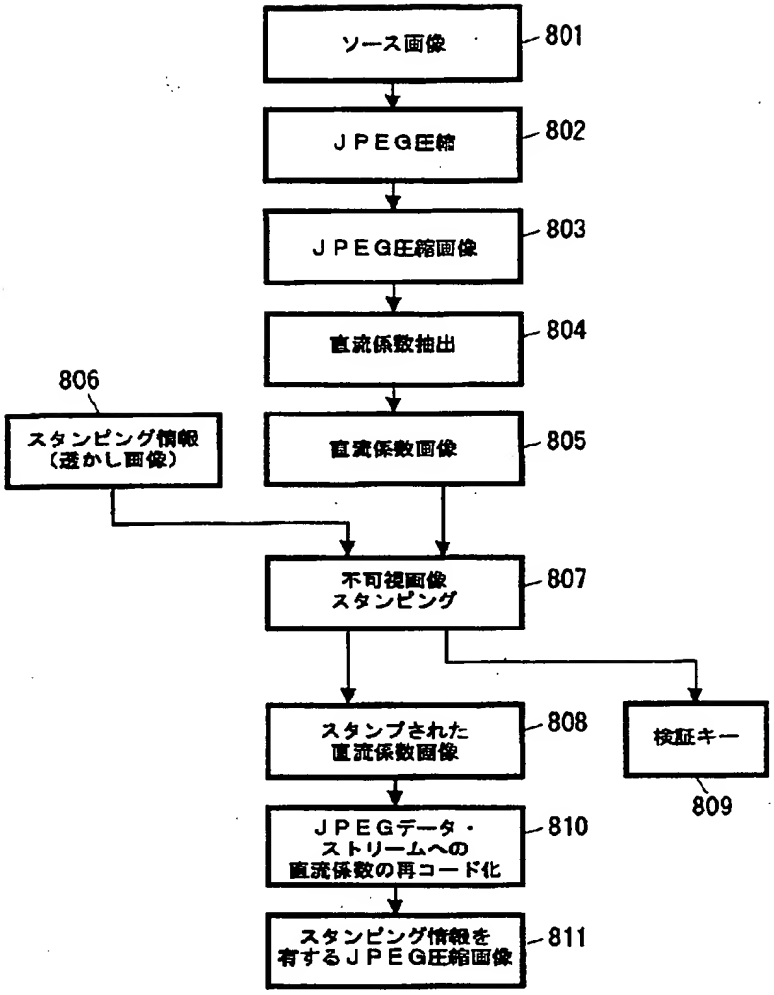
【図１０】


SS(I，J)
R
G
B
入力画素
1001
SS_R(I，J)
SS_G(I，J)
SS_B(I，J)
テーブル
1002
LUT_R
LUT_G
LUT_B
V1
V2
V3
XOR
1003
1004
EW(I，J)
1005

【図7】


修正誤差値を計算する
MP(I, J) =S(I, J)+E(I, J)
701
透かし値を抽出する
EW=WX(MP(I, J))
702
703
EW=
W(I, J)か
?
YES
EW=W(I, J)
NO
EW≠W(I, J)
705
Δを初期設定する
704
SS(I, J) =
MP(I, J)
706
新しいΔを計算する
707
修正画素値MSを変更する
MS=MP(I, J) +Δ
708
透かし値EMWをMSから抽出する
EMW=WX(MS)
709
EMW=
WX(MS)
か?
NO
EMW≠W(I, J)
YES
EMW=W(I, J)
710
SS(I, J) =MS
711
E(I, J) =
MP(I, J) -S(I, J)
712
誤差拡散 D(I, J)
終了
713

【図８】



フロントページの続き

(72)発明者　マナーグァ・ミン＝イー・ユン
アメリカ合衆国10598　ニューヨーク州ヨークタウン・ハイツ　ハイツ・ドライブ
1537